JOINT TECHNICAL REPORT

"ALBERT"

A PACKAGE OF FOUR COMPUTER PROGRAMS FOR CALCULATING GENERAL RELATIVISTIC CURVATURE TENSORS AND EQUATIONS OF MOTION*

KIP S. THORNE† and BARBARA A. ZIMMERMAN

California Institute of Technology, Pasadena, California

ABSTRACT

This report is a user's manual for ALBERT, a package of four Caltech computer programs which calculate analytic expressions for general relativistic curvature tensors and equations of motion. The input data for ALBERT are analytic forms for the covariant and contravariant components of the metric tensor, g_{ab} and g^{ab}, and for the mixed components of the stress-energy tensor, $W_a{}^b$. The output, variable at the discretion of the user, includes analytic expressions for the equations of motion, $W_a{}^b{}_{;b} = 0$; for the Christoffel symbols, Γ_{abc} and $\Gamma_{ab}{}^c$; for the Riemann tensor, R_{abcd}; for the Ricci tensor, $R^a{}_b$; for the Einstein tensor, $G^a{}_b$; and for the Ricci scalar, R. An option is included for expanding the answers in a power series in some parameter, ϵ, which appears in the input tensors; and for keeping only terms of zero, first, and second order. ALBERT is written in the IBM language FORMAC, which is quite similar to FORTRAN; but which, unlike FORTRAN, provides for algebraic manipulation of symbols and for analytic differentiation of functions. ALBERT should be usable, without modification, on any computer which has a FORMAC compiler. A complete listing of the programs in ALBERT is included in this report.

*Supported in part by the National Science Foundation [GP-5391] and the Office of Naval Research [Nonr-220(47)].

†Alfred P. Sloan Foundation Research Fellow.

THIS TECHNICAL REPORT IS ONE OF THE GRAPE AID SERIES OF SCIENTIFIC MEMORANDA, TECHNICAL REPORTS & MISCELLANEOUS WRITINGS.

August 1967

82

Blank Page

TABLE OF CONTENTS

Page

I. Introduction and Motivation 1

*II. Brief Description of ALBERT 2

III. How to Use the Programs in ALBERT 4

*A. Brief Description of the FORMAC Language 4

*B. EINSTEIN, the Program for Calculating Ricci and Einstein Tensors 9

1. Input for EINSTEIN 9

2. Complete Listing of EINSTEIN 13

3. Output for EINSTEIN 21

4. Sample Problems for EINSTEIN 25

C. RIEMANN, the Program for Calculating Riemann Tensors . . . 34

1. Input for RIEMANN 34

2. Complete Listing of RIEMANN 35

3. Output for RIEMANN 40

4. Sample Problems for RIEMANN 41

D. MOTION, the Program for Calculating the Iivergence of the Stress-Energy Tensor 42

1. Input for MOTION 42

2. Complete Listing of MOTION 43

3. Output for MOTION 49

4. Sample Problems for MOTION 51

E. SUBSTITUTE, the Program for Making Changes of Variables . . 56

1. Input for SUBSTITUTE 56

*The reader interested in setting up, as rapidly as possible, a working program for calculating Einstein and Ricci tensors need read only these sections.

Page

2. Complete Listing of SUBSTITUTE 59

3. Output of SUBSTITUTE 62

4. Sample Problems for SUBSTITUTE 64

IV. A Brief Guide to FORMAC Programming 69

A. The FORMAC System 69

B. FORMAC Executable Statements 70

C. Ways to Avoid Exhausting the Computer's Memory 75

V. References 76

I. INTRODUCTION AND MOTIVATION

The initial step in many general-relativity research projects is to select a coordinate system and corresponding analytic forms for the metric tensor, g_{ab}, and the stress-energy tensor, $W_a{}^b$; and then to compute the equations of motion, $W_a{}^b{}_{;b} = 0$, and the Einstein tensor, $G_a{}^b$, for use in the field equations $G_a{}^b = 8\pi W_a{}^b$. At later stages, one also has occasional need for the Riemann and Ricci tensors, R_{abcd} and $R^a{}_b$, and for the Ricci scalar, R.

So long as one deals with simple systems which possess a high degree of symmetry, the computation by hand of $W_a{}^b{}_{;b}$, $G_a{}^b$, R_{abcd}, $R^a{}_b$, and R is not too tedious. However, in recent years, the systems under study have become more and more complex; and the task of computing the necessary equations of motion and curvature tensors has become more and more formidable. Recent computations of curvature tensors have sometimes required many weeks of tedious labor, and the possibilities of mistakes of minus signs and factors 2 in the final results have been very disturbing.

In response to this situation, we have developed the computer programs described here for calculating curvature tensors and equations of motion. These programs have been used extensively over the last seven months in relativistic astrophysics research at Caltech. (See, e.g., the analysis of the effects of rotation on the structure of general relativistic stellar models by Hartle (1967); also, the analysis of nonradial pulsations of relativistic stellar models by Thorne and Campolattaro (1967).)

After these programs were completed and in use at Caltech, we learned that Fletcher (1965, 1966) and Clemens and Matzner (1967) have developed independently computer programs for doing the same job. The three different systems are describeo ·riefly in Fletcher, Clemens, Matzner, Thorne, and Zimmerman (1967).

II. BRIEF DESCRIPTION OF ALBERT

ALBERT is a package consisting of four independent computer programs -- EINSTEIN, RIEMANN, MOTION, and SUBSTITUTE -- which perform four independent tasks.

EINSTEIN is a program which takes a metric tensor, g_{ab}, and calculates for it the Christoffel symbols, Γ_{abc} and $\Gamma_{ab}{}^{c}$, the mixed components of the Ricci tensor, $R_a{}^b$, the Ricci scalar, R, and the mixed components of the Einstein tensor, $G_a{}^b = R_a{}^b - \frac{1}{2} R \delta_a{}^b$. The input to EINSTEIN are functional forms for g_{ab} and g^{ab}; and the output are printed expressions for all the above quantities.

RIEMANN is a program for calculating the covariant components, R_{abcd}, of the Riemann curvature tensor for any given metric tensor, g_{ab}. The input to RIEMANN are functional forms for g_{ab} and g^{ab}; and the output are printed expressions for Γ_{abc}, $\Gamma_{ab}{}^{c}$, and R_{abcd}.

MOTION is a program for calculating the divergence, $W_a{}^b{}_{;b}$, of a stress-energy tensor, $W_a{}^b$. (We denote the stress-energy tensor by $W_a{}^b$ rather than $T_a{}^b$ because the symbol T is reserved for the time coordinate.) The input to MOTION are functional forms for g_{ab}, g^{ab}, and $W_a{}^b$; and the output are printed expressions for Γ_{abc}, $\Gamma_{ab}{}^{c}$, and $W_a{}^b{}_{;b}$.

SUBSTITUTE is a program for making changes of variables in long, complicated expressions. The input to SUBSTITUTE are the changes of variables desired, e.g.

$$\lambda = - \ln (1-2m/X);$$

and the expressions in which those changes are to be made, e.g.

$$G_4{}^4 = e^{-\lambda} \left(- \frac{1}{X^2} + \frac{\partial\lambda/\partial X}{X} \right) + \frac{1}{X^2} .$$

The output are printed expressions for the desired quantities in terms of the new variables, e.g.

$$G_4{}^4 = 2 \frac{\partial m/\partial X}{X^2} .$$

The programs EINSTEIN, RIEMANN and MOTION are severely limited by core storage problems: They are incapable of calculating curvature tensors and equations of motion for metrics and stress-energy tensors which are too complicated. (Examples of input which will and will not compute are given in §§ III.B.4, III.C.4, III.D.4, and III.E.4.) However, complicated problems can be solved by putting into EINSTEIN, RIEMANN, and MOTION simplified metric forms, e.g.

$$g_{44} = A(X,Y,T)$$

rather than

$$g_{44} = e^{\nu(x)} \{1. + A_1(X,T)\ P_1(\cos Y) + A_2(X,T)\ P_2(\cos Y)\};$$

and by then using SUBSTITUTE to make a change of variables in the resultant curvature tensors and equations of motion, e.g.

$$A(X,Y,T) = e^{\nu(x)} \{1. + A_1(X,T)\ P_1(\cos Y) + A_2(X,T)\ P_2(\cos Y)\}.$$

An option is included in EINSTEIN, RIEMANN, MOTION, and SUBSTITUTE for expanding the output in powers of a small parameter and retaining only terms of zero, first, and second order.

EINSTEIN, RIEMANN, MOTION, and SUBSTITUTE are all written in the IBM language FORMAC. FORMAC is an extension of FORTRAN IV, but unlike FORTRAN IV, FORMAC can currently be used only on IBM 7090 and 7094 computers. Hopefully an improved version of FORMAC will be available for the IBM 360 system sometime in 1968.

III. HOW TO USE THE PROGRAMS IN ALBERT

A. Brief Description of the FORMAC Language

In this section we present only enough information on FORMAC to enable the reader to use the programs in ALBERT. A more detailed description of FORMAC, intended to help the reader who wishes to write simple FORMAC programs for himself or to modify ALBERT to suit his own individual needs, will be found in § IV of this report. For a complete description of the FORMAC system and for guidance in writing complicated FORMAC programs, the reader is referred to the standard IBM FORMAC manual (IBM 1964).

The FORMAC system is capable of performing formal algebraic manipulations, and differentiations of algebraic expressions. It can also evaluate numerically the algebraic expressions which it creates, but no such numerical evaluations are used in the programs described in this report.

FORMAC uses notation which is almost identical to that of FORTRAN IV:

1. All algebraic quantities -- constants, parameters, spacetime coordinates, functions, algebraic expressions -- are represented by "words" which are constructed from 6 or fewer Latin letters and digits. For example, in ALBERT we use X,Y,Z, and T to represent the 4 spacetime coordinates; and we use EP to represent a small parameter in terms of which power series expansions are made. As further examples, in the sample problem given in § III.B.4, ANU represents the function $\nu(X)$ and H1 represents the function $H_1(X,T)$. As a final example, with the above definitions of X, Y, Z, T, EP, ANU, and H1, the statement

   LET PAM3 = T * ANU + H1/Y,

   tells the computer to use the word PAM3 to represent the algebraic expression T $\nu(X) + H_1(X,T)/Y$. (When defining an algebraic expression, one must always use the word LET in front of the definition. Otherwise, the computer will attempt to evaluate the expression numerically, in the usual FORTRAN manner.)

2. Algebraic quantities can be subscripted in the same manner as is done in FORTRAN. For example, we denote the covariant components of the metric tensor, g_{ij}, by the words Gl(I,J); Gl(1,3) is g_{13}, Gl(2,2) is g_{22}, etc.

3. The computer distinguishes constants and parameters from functions and from algebraic expressions by means of "ATØMIC" and "DEPEND" statements, which appear near the beginning of the program. (These statements are similar to the EXTERNAL statement of FORTRAN IV.)

   a. The ATØMIC statement identifies all algebraic quantities ("atomic variables") which do <u>not</u> represent internally constructed algebraic expressions. All nonnumerical constants, parameters, spacetime coordinates, and functions must appear in the ATØMIC statement. For example, for a program involving only the quantities described in 1 above, all quantities except PAM3 appear in the ATØMIC statement, and the atomic statement takes the form:

      ATØMIC X, Y, Z, T, EP, ANU, H1.

   b. The DEPEND statement identifies all <u>functions</u> which are <u>not</u> algebraic expressions -- i.e., which are atomic --; and it states the functional dependences of the functions. For example, one identifies $\nu(X)$ and $H_1(X,T)$ as atomic functions by the above ATØMIC statement plus the DEPEND statement

      DEPEND (ANU/X), (H1/X,T).

   Note that PAM3 as defined above is <u>not</u> an atomic function; it is an algebraic expression made up of atomic functions and other atomic variables, and it therefore does not appear in either an ATØMIC statement or a DEPEND statement.

4. In constructing algebraic expressions one can perform the elementary operations of addition, subtraction, multiplication, division, and exponentiation (i.e., raising to a power). The notation for these operations is identical to that of FORTRAN IV:

+ denotes addition

- denotes subtraction

* denotes multiplication

/ denotes division

** denotes exponentiation.

As in FORTRAN IV, the order in which these operations are performed is **, followed by * and /, followed by + and -. For example, consider the following FORMAC statements and their translation to ordinary algebraic language:

FORMAC LANGUAGE	CONVENTIONAL NOTATION
LET PAM3 = T*ANU*H1/Y	this sets PAM3 equal to $\frac{T\nu H_1}{Y}$
LET SS = PAM3**2.+3.*(X-Y)**T	this sets SS equal to $\left(\frac{T\nu H_1}{Y}\right)^2 + 3(X-Y)^T$

In constructing algebraic expressions one can use numbers which are floating point (i.e. have decimals like 2. and 3. above) or which are integers (e.g. 2 and 3). When doing floating point manipulations the computer generally outputs exponential notation -- e.g. "2.5E-1" ($2.5\text{x}10^{-1}$) instead of 1/4. When doing integer algebraic manipulations, FORMAC uses 2**(-2) instead of 2.5E-1. When integer (i.e. rational) expressions are mixed together with floating point expressions by the user, the computer converts them all to floating point before manipulating.

5. The computer recognizes and knows how to differentiate the following elementary functions:

FORMAC LANGUAGE	CONVENTIONAL NOTATION
FMCEXP(ANU)	exponential: e^{ν}
FMCSIN(2.*T)	sine: sin(2T)
FMCCØS(H1**2)	cosine: $\cos(H_1{}^2)$
FMCLØG(Y)	natural logarithm: log(Y)
FMCATN(FMCEXP(ANU))	arctangent: $\tan^{-1}(e^{\nu})$
FMCHTN(3*Z+7)	hyperbolic tangent: tanh(3Z+7)

These functions can be used freely in algebraic expressions.

6. FORMAC is capable of differentiating any algebraic expression constructed according to the above rules. If A is a function of X, Y, T, but not of Z

DEPEND (A/X,Y,T),

then various derivatives of A are denoted as follows

FORMAC LANGUAGE	CONVENTIONAL NOTATION
FMCDIF(A,(X,1))	$\partial A/\partial X$
FMCDIF(A,(X,2))	$\partial^2 A/\partial X^2$
FMCDIF(A,(X,1),(Y,1))	$\partial^2 A/\partial X \partial Y$
FMCDIF(A,(X,2),(T,1))	$\partial^3 A/\partial X^2 \partial T$

In constructing algebraic expressions, one can use FORMAC differentiation freely, and the requested differentiation will always be performed to completion. For example, consider the following sequence of commands:

FORMAC COMMAND	RESULT OF COMMAND
ATØMIC X, Y, Z, T, EP, ANU, H1	Defines the atomic variables for the program.
DEPEND (ANU/X),(H1/X,T)	Defines the atomic functions and their dependences.
LET PAM3 = T*ANU*H1/Y	Sets PAM3 equal to $T\nu H_1/Y$.
LET Q1 = FMCDIF(PAM3,T,1)	Performs the required differentiation on PAM3, yielding $Q1 = \nu H_1/Y + (T\nu/Y)\partial H_1/\partial T.$
LET CC = FMCDIF(FMCSIN(H1)+T*X, (T,1),(X,1))	Performs the required differentiation yielding $CC = (\cos H_1)(\partial^2 H_1/\partial T\partial X)$ $- (\sin H_1)(\partial H_1/\partial T)(\partial H_1/\partial X)$ $+ 1$
LET CD = FMCDIF(ANU,Y)	Performs the required differentiation yielding $CD = 0.$

There are commands in the FORMAC language which remove parentheses from algebraic expressions, which factor terms out thereby inserting parentheses, which truncate power series, and which perform other useful manipulations. However, none of these commands are necessary to the use of the programs in ALBERT, so we shall delay describing them until § IV.B.

B. EINSTEIN, the Program for Calculating Ricci and Einstein Tensors

1. Input for EINSTEIN

EINSTEIN is the program which calculates Ricci and Einstein tensors. In order to use EINSTEIN, one must supply the following:

1. An ATØMIC statement identifying all atomic variables for the problem. The ATØMIC statement takes the form

   ATØMIC X, Y, Z, T, EP, ...

   The user inserts after EP all functions and parameters which appear in his metric tensor.

2. A DEPEND statement identifying all atomic functions which appear in the metric tensor and specifying their functional dependences. For the form of the DEPEND statement, see page 5.

3. A statement declaring the value of the integer EPTERM. EPTERM is used to tell the computer whether or not to expand the curvature tensors to second order in the parameter EP and truncate thereafter (i.e., keep only terms in EP**0, EP**1, and EP**2).

   If EPTERM = 0 terms in all powers of EP will be retained.

   If EPTERM ≠ 0 only terms of order 0, 1, 2 in EP will be retained.

4. "LET" statements defining all nonzero covariant components of the metric tensor, g_{ij}, in terms of atomic variables. g_{ij} is denoted in ALBERT by G1(I,J). The metric components 1, 2, 3, 4 refer to the spacetime coordinates X, Y, Z, T respectively.

5. "LET" statements defining all nonzero contravariant components of the metric tensor, g^{ij}, in terms of atomic variables. g^{ij} is denoted in ALBERT by G2(I,J). The matrix G2(I,J) must be, of course, the algebraic inverse of G1(I,J) – at least up to second order in the expansion parameter EP.

These 5 statements are inserted by the user directly into the body of the program EINSTEIN at the points dilineated by comment cards. (See listing in the next section).

For example (another example is given in § III.B.4), to calculate Ricci and Einstein tensors for the standard, static, spherically symmetric metric

$$ds^2 = -e^{\lambda(r)}dr^2 - r^2(d\theta^2 + \sin^2\theta\, d\varphi^2) + e^{\nu(r)}dt^2 \tag{1}$$

one must:

1. Mentally replace r by X, θ by Y, φ by Z, t by T, ν by ANU, and λ by ALAM;
2. Supply the following cards at the required points in EINSTEIN

```
ATØMIC X, Y, Z, T, EP, ANU, ALAM
DEPEND (ANU,ALAM/X)
EPTERM = 0
LET G1(1,1) = -FMCEXP(ALAM)
LET G1(2,2) = -X**2
LET G1(3,3) = -(X*FMCSIN(Y))**2
LET G1(4,4) = FMCEXP(ANU)
LET G2(1,1) = -FMCEXP(-ALAM)
LET G2(2,2) = -1/X**2
LET G2(3,3) = -1/(X*FMCSIN(Y))**2
LET G2(4,4) = FMCEXP(-ANU)
```

As in FORTRAN all cards supplied by the user must have their first entry in or after column 7 and their last entry in or before column 72. Long statements can be continued over as many as nineteen continuation cards. Any card with a nonzero entry in column 6 is considered a continuation card.

3. Run the program on an IBM-7090 or 7094 computer using a IBSYS-FORMAC operational tape (obtainable from IBM), which contains the FORMAC compiler.

The user is free to give the functions and variables entering into his metric tensor any names constructed from 6 or fewer characters, except the following:

NAMES WHICH MUST NOT BE USED AS

VARIABLE NAMES IN ALBERT

1. X, Y, Z, T <u>except</u> for use as spacetime coordinates.
2. EP <u>except</u> for use as a constant (not functional) expansion parameter; and EPTERM, <u>except</u> for use to control expansions in EP.
3. G1 or G2 <u>except</u> for use as metric tensors.
4. W <u>except</u> for use as stress-energy tensor.
5. Any name beginning with a digit (0, ..., 9); i.e., all names must begin with Latin letters.
6. Any name <u>beginning with</u> the following (these are forbidden in all FØRMAC programs):

ALGCØN
ATØMIC
AUTSIM
BCDCØN
CENSUS
CØMPLE
CØDEM
CØEFF
DEPEND
DIMENS

DØUBLE
END
ERASE
EVAL
EXPAND
FIND
FMC
FUNCTI
IMC
INTEGE
LET

LØGICA
MATCH
ØRDER
PARAM
PART
REAL
SUBST
SUBRØU
SYMARG

7. The following names, which are used for various other purposes by ALBERT

ANS, AA, AAA
BEGIN, BLANK
CARD, CCS
D1, D2, DIFF
EM, EXP
I, J, K, L, M, N
II, JJ, JJJ, LL
LINE, LBL
MCS, MU, NU
NSUBS, NEXP
PP, Q, R
RICCI, RTEN, RMT
S, S0, S1, ..., S5
SCAL
SUM
T1, T2, T3, TT
Z0, Z1, Z2

FORMAC requires a huge amount of core storage for its operations. Consequently, if one inputs metric tensors which are too complicated, he will exhaust the memory capacity of the computer before finishing the computation. When this happens, the only way of using the ALBERT programs as they stand is to simplify the form of the metric tensor. The simplified Ricci and Einstein tensors which result therefrom can be reexpressed in terms of the more complicated metric by means of the program SUBSTITUTE. For examples of metrics which did and did not compute without simplification, and of the simplifications used for certain complicated metrics, see §§ III.B.4, III.C.4, III.D.4, and III.E.4.

One of the most effective ways to avoid exhausting the computer's memory is to avoid using complicated denominators in the expressions for the metric tensor. FORMAC does not handle denominators efficiently. For example

G1(1,1) = 1./(1.-2*EM/X),

where EM depends on X, is much more likely to produce trouble than

G1(1,1) = FMCEXP(ALAM),

where ALAM depends on X.

2. Complete Listing of EINSTEIN

In the listing of Einstein which follows, and in all other program listings in this report, all control cards are omitted. (Control cards are cards which precede and follow each program and which tell the computer such things as who the user of the program is and what language the program is written in.) Because control cards vary in format from one computer installation to another, the user should consult his friendly computer representative for special instructions on their use. The user should warn his computer representative that he is using FORMAC, which requires somewhat different control cards than other languages.

```
$IBFMC ONE     NODECK
         SYMARG
C
C          EINSTEIN, A PROGRAM TO CALCULATE THE RICCI AND EINSTEIN TENSORS
C
       ATOMIC  X,Y,Z,T,EP,ANU,ALAM          }  User:  Replace these cards
       DEPEND (ANU,ALAM/X)                  }  with your own input.
C
C
C         ATOMIC AND DEPEND STATEMENTS MUST BE SUPPLIED BY USER
C         IMMEDIATELY PRECEDING THIS COMMENT.
C         THE ATOMIC STATEMENT MUST LIST ALL FUNCTIONS AND INDEPENDENT
C         VARIABLES USED IN A PARTICULAR PROBLEM.
C         THE DEPEND STATEMENT DEFINES ALL THE FUNCTIONAL DEPENDENCIES
C         FOR THE PROBLEM, I.E. WHICH FUNCTIONS DEPEND ON WHICH VARIABLES.
C
        REAL MCS(4,4,4)
        LOGICAL Q
        DIMENSION CARD(12),LINE(12),CCS(4,4,4),G1(4,4),G2(4,4),II(4)
       DIMENSION RICCI(4)
C
        INTEGER EPTERM
C
C         IF EPTERM = 0 TERMS CONTAINING ALL POWERS OF EXPANSION
C                       PARAMETER WILL BE RETAINED.
C         IF EPTERM .NE. 0 TERMS CONTAINING THE EXPANSION PARAMETER
C                       TO A POWER GREATER THAN 2 WILL BE DISCARDED.
C        A CARD CONTAINING THE INTEGER VALUE OF EPTERM MUST FOLLOW
C                       THIS COMMENT.
C
       EPTERM = 0                            }  User:  Replace this card
C                                                with your own input.
        DO 1 I = 1,4
        DO 1 J = 1,4
        LET G1(I,J) = 0.
        LET G2(I,J) = 0.
    1  CONTINUE
C
C        ALL ELEMENTS OF THE COVARIANT METRIC TENSOR, G1(I,J), AND THE
C        CONTRAVARIANT METRIC TENSOR, G2(I,J), HAVE BEEN SET EQUAL
C        TO ZERO.
C        USER MUST SUPPLY ''LET'' STATEMENTS DEFINING ALL NON ZERO
C        ELEMENTS OF THESE TENSORS IMMEDIATELY FOLLOWING THIS COMMENT.
C
C
      LET G1(1,1) = -FMCEXP(ALAM)              }
      LET G1(2,2) = -X**2                      }
      LET G1(3,3) = -(X*FMCSIN(Y))**2          }
      LET G1(4,4) =  FMCEXP(ANU)               }  User:  Replace these cards
      LET G2(1,1) = -FMCEXP(-ALAM)             }  with your own input.
      LET G2(2,2) = -1/X**2                    }
      LET G2(3,3) = -1/(X*FMCSIN(Y))**2        }
      LET G2(4,4) =  FMCEXP(-ANU)              }
```

```
C
C
C     THE COVARIANT METRIC TENSOR -- OUTPUT
C
       WRITE(6,503)
  503 FORMAT( 51H1COVARIANT METRIC TENSOR NON ZERO ELEMENTS  (INPUT)  )
       DO 2 I = 1,4
       DO 2 J=1,4
       LET Q = MATCH ID, G1(I,J),0.
       IF(Q) GO TO 2
       WRITE(6,800)I,J
  800 FORMAT(1H I5,1HL I5,1HL )
       BEGIN = 0.
   46 LET BEGIN = BCDCON G1(I,J),LINE,12
       WRITE(6,200)(LINE(L),L=2,12)
       IF(BEGIN.NE.0.)GO TO 46
    2 CONTINUE
C
C     THE CONTRAVARIANT METRIC TENSOR  -- OUTPUT
C
      WRITE(6,700)
  700 FORMAT(54H0CONTRAVARIANT METRIC TENSOR NON ZERO ELEMENTS (INPUT))
       DO 47 I = 1,4
       DO 47 J=1,4
       LET Q = MATCH ID, G2(I,J),C.
       IF(Q)GO TO 47
      WRITE(6,500) I,J
  500 FORMAT(1H I5,1HU I5,1HU )
      BEGIN = 0.
   48 LET BEGIN = BCDCON G2(I,J),LINE,12
       WRITE(6,200)(LINE(L),L=2,12)
       IF(BEGIN.NE.0.)GO TO 48
   47 CONTINUE
C
C     THE COVARIANT CHRISTOFFEL SYMBOLS
C
       WRITE(6,201)
  201 FORMAT(34H1THE COVARIANT CHRISTOFFEL SYMBOLS )
       DO 3 I = 1,4
       DO 3 J = I,4
       DO 3 K = 1,4
       LET T1 = 0.
       GO TO (31,32,70,71),I
   31 LET T1 = FMCDIF(G1(J,K),X,1)
       GO TO 33
   32 LET T1 = FMCDIF(G1(J,K),Y,1)
       GO TO 33
   70 LET T1 = FMCDIF(G1(J,K),Z,1)
       GO TO 33
   71  LET T1  = FMCDIF(G1(J,K),T,1)
   33  CONTINUE
        LET T2 = 0.
        GO TO(35,36,72,73),J
   35  LET T2  = FMCDIF(G1(I,K),X,1)
        GO TO 34
   36  LET T2  = FMCDIF(G1(I,K),Y,1)
         GO TO 34
   72  LET T2  = FMCDIF(G1(I,K),Z,1)
         GO TO 34
   73  LET T2  = FMCDIF(G1(I,K),T,1)
   34  CONTINUE
        LET T3 = 0.
        GO TO (38,39,74,75),K
   38  LET T3  = FMCDIF(G1(I,J),X,1)
        GO TO 37
   39  LET T3  = FMCDIF(G1(I,J),Y,1)
        GO TO 37
   74  LET T3  = FMCDIF(G1(I,J),Z,1)
        GO TO 37
   75  LET T3  = FMCDIF(G1(I,J),T,1)
   37  LET CCS(I,J,K)  = 0.5*(T1+T2-T3)
        LET CCS(J,I,K)  = CCS(I,J,K)
        LET Q  = MATCH ID, CCS(I,J,K),0.
        IF(Q)GO TO 3
        WRITE(6,203)I,J,K
  203  FORMAT(1H I5,2HLS I3,2HLS  I4,1HL)
        LET ANS  = EXPAND CCS(I,J,K)
        LET ANS  = ORDER ANS,INC,FUL
        BEGIN  = 0.
    5  LET BEGIN  = BCDCON ANS,LINE,12
        WRITE(6,200)(LINE(L),L=2,12)
  200  FORMAT(1H   5X 12A6)
        IF(BEGIN.NE.0.)GO TO 5
        ERASE  ANS
    3  CONTINUE
         ERASE  T1,T2,T3
C
C        THE  MIXED  CHRISTOFFEL  SYMBOLS
C
        WRITE(6,205)
  205  FORMAT( 30H1THE MIXED CHRISTOFFEL  SYMBOLS  )
        DO 6  I  = 1,4
        DO 6  J  = I,4
        DO 6  K  = 1,4
        LET MCS(I,J,K)  = 0.
        DO 7  L  = 1,4
        LET TT  = G2(K,L)*CCS(I,J,L)
        LET TT  = EXPAND TT
        IF(EPTERM)506,507,506
  507  LET  SUM  = TT
```

```
        ERASE TT
        GO TO 514
  506  CONTINUE
        LET S1 = COEFF TT,EP**0,R
        LET S2 = COEFF TT,EP**1,R
        LET S3 = COEFF TT,EP**2,R
        ERASE TT
        LET SUM = S1 + S2*EP + S3*EP*EP
       ERASE S1,S2,S3
 514   CONTINUE
        LET MCS(I,J,K) = MCS(I,J,K) + SUM
        ERASE SUM
    7  CONTINUE
        LET MCS(I,J,K) = EXPAND MCS(I,J,K)
        LET MCS(J,I,K) = MCS(I,J,K)
        LET Q = MATCH ID, MCS(I,J,K),0.
        IF(Q)GO TO 6
       WRITE(6,769) I,J,K
  769  FORMAT(1H I5,2HLS I3,2HLS I4,1HU )
        LET ANS = ORDER MCS(I,J,K),INC,FUL
        BEGIN = 0.
    8  LET BEGIN = BCDCON ANS,LINE,12
        WRITE(6,200)(LINE(L),L=2,12)
        IF(BEGIN.NE.0.)GO TO 8
        ERASE ANS
    6  CONTINUE
C
C
       DO 350 I=1,4
       DO 350 J=1,4
       DO 350 K=1,4
       ERASE CCS(I,J,K)
  350 CONTINUE
C         THE RICCI TENSOR
C
        WRITE(6,217)
  217  FORMAT(36H1THE RICCI TENSOR, MIXED COMPONENTS    )
        DO 50 N = 1,4
        DO 50 K = 1,4
        LET S1 = 0.
        DO 51 I = 1,4
        DO 51 M = 1,4
        LET T1 = 0.
        GO TO (21,22,23,24),I
   21  LET T1 = FMCDIF(MCS(N,M,I),X,1)
        GO TO 19
   22  LET T1 = FMCDIF(MCS(N,M,I),Y,1)
        GO TO 19
   23  LET T1 = FMCDIF(MCS(N,M,I),Z,1)
        GO TO 19
   24  LET T1 = FMCDIF(MCS(N,M,I),T,1)
 19   CONTINUE
       LET T1 = EXPAND T1
       LET T2 = 0.
       GO TO (52,53,54,55),N
 52   LET T2 = -FMCDIF(MCS(I,M,I),X,1)
       GO TO 56
 53   LET T2 = -FMCDIF(MCS(I,M,I),Y,1)
       GO TO 56
 54   LET T2 = -FMCDIF(MCS(I,M,I),Z,1)
       GO TO 56
 55   LET T2 = -FMCDIF(MCS(I,M,I),T,1)
 56   CONTINUE
       LET T2 = EXPAND T2
       LET TT = G2(K,M)*(T1+T2)
        ERASE T1,T2
       LET TT = EXPAND  T1
       IF(EPTERM)504,513,504
513   LET S1 = S1 + TT
       ERASE TT
       LET S1 = EXPAND S1
       GO TO 51
504   CONTINUE
       LET S0 = COEFF TT,EP**0,R
       LET S3 = COEFF TT,EP**1,R
       LET S4 = COEFF TT,EP**2,R
       ERASE TT
       LET S1 = S1 + (S0 + S3*EP + S4*EP*EP)
       ERASE S0,S3,S4
       LET S1 = EXPAND S1
 51   CONTINUE
       LET S2 = 0.
       DO 84 I = 1,4
       DO 84 J = 1,4
       DO 84 M = 1,4
      LET TT   =    G2(K,M)*(-MCS(J,M,I)*MCS(I,N,J)+MCS(M,N,I)*
    1 MCS(I,J,J))
       LET TT = EXPAND  T1
       IF(EPTERM)510,511,510
511   LET S2 = S2 + TT
       ERASE TT
       LET S2 = EXPAND S2
       GO TO 84
510   CONTINUE
       LET S0 = COEFF TT,EP**0,R
       LET S3 = COEFF TT,EP**1,R
       LET S4 = COEFF TT,EP**2,R
       ERASE TT
       LET S2 = S2 + (S0 + S3*EP + S4*EP*EP)
       ERASE S0,S3,S4
       LET S2 = EXPAND S2
 84   CONTINUE
```

```
       LET RTEN = S1 + S2
       ERASE S1,S2
       LET RTEN= EXPAND RTEN
       IF (N-K) 983,984,983
 984   LET RICCI(N) = RTEN
 983   CONTINUE
       LET Q = MATCH ID,RTEN,0.
       IF(Q)GO TO 50
       LET ANS = ORDER RTEN,INC,FUL
        WRITE(6,216)K,N
  216 FORMAT(1H I5,1HU I5,1HL  )
  501 FORMAT(2I5)
       BEGIN = 0.
   58  LET BEGIN = BCDCON ANS,LINE,12
       WRITE(6,200)(LINE(IL),IL = 2,12)
  502 FORMAT(12A6)
       IF(BEGIN.NE.0.)GO TO 58
       ERASE ANS
   50 CONTINUE
   92 CONTINUE
C
C      THE RICCI SCALAR
C
       LET SCAL = RICCI(1) + RICCI(2)
       LET SCAL = EXPAND SCAL
       LET SCAL = SCAL + RICCI(3)
       LET SCAL = EXPAND SCAL
       LET SCAL = SCAL + RICCI(4)
       LET SCAL = EXPAND SCAL
       LET ANS = ORDER SCAL,INC,FUL
       WRITE (6,233)
 233   FORMAT (17H1THE RICCI SCALAR)
       BEGIN = 0.
 234  LET BEGIN = BCDCON ANS,LINE,12
       WRITE(6,200)(LINE(IL),IL=2,12)
       IF (BEGIN.NE.0.) GO TO 234
       ERASE ANS
C
C      THE EINSTEIN TENSOR, DIAGONAL COMPONENTS
C
      WRITE(6,221)
  221 FORMAT( 58H1THE EINSTEIN TENSOR, MIXED COMPONENTS, DIAGONAL ELEMEN
     1TS      )
      DO 516 L=1,4
       LET ANS = RICCI(L) - 0.5*SCAL
       LET ANS = EXPAND ANS
      LET ANS = ORDER ANS, INC, FUL
       WRITE(6,219)L,L
  219 FORMAT(1H I5,1HU I5,1HL)
      BEGIN = 0.
  342 LET BEGIN = BCDCON ANS, LINE, 12
```

```
      WRITE(6,200) (LINE(IL),IL=2,12)
      IF(BEGIN.NE.0.) GO TO 342
      ERASE ANS
  516 CONTINUE
   93 CONTINUE
      STOP
      END
```

3. Output for EINSTEIN

The printed output for EINSTEIN includes: the metric components, g_{ij} and g^{ij}, which were input by the user; the Christoffel symbols, which are calculated from the formulas

$$\Gamma_{ijk} = \frac{1}{2} (g_{ik,j} + g_{jk,i} - g_{ij,k}), \qquad (2)$$

$$\Gamma_{ij}{}^{m} = g^{mk} \Gamma_{ijk} , \qquad (3)$$

(note that the Christoffel symbols are symmetric in the first two indices); the mixed components of the Ricci curvature tensor, calculated from

$$R^{k}{}_{n} = g^{km} \left\{ \Gamma_{mn}{}^{i}{}_{,i} - \Gamma_{im}{}^{i}{}_{,n} + \Gamma_{mn}{}^{i} \Gamma_{ij}{}^{j} - \Gamma_{jm}{}^{i} \Gamma_{in}{}^{j} \right\} ; \qquad (4)$$

the Ricci scalar,

$$R = R^{k}{}_{k} ; \qquad (5)$$

and the diagonal mixed components of the Einstein tensor,

$$G^{k}{}_{n} = R^{k}{}_{n} - \frac{1}{2} R \, \delta^{k}{}_{n} . \qquad (6)$$

(Recall that the off-diagonal components of the Einstein tensor are identical to the off-diagonal components of the Ricci tensor.) Components which are zero are not printed out; and when two components are equal by symmetry considerations (e.g., $\Gamma_{123} = \Gamma_{213}$), only one of them is printed.

The format of the printed output is self-explanatory except for the following: Before each component of a quantity the computer prints the numerical values of the indices, plus symbols to identify whether the indices are contravariant (U for upper) or covariant (L for lower). For example, in the Ricci tensor

$$1U\ 3L \leftrightarrow R^{1}_{3}\ .$$

Moreover, in the Christoffel symbols an S is used to identify the indices which are symmetric:

$$1LS\ 3LS\ 4L \leftrightarrow \Gamma_{134},$$ symmetric under interchange of 1 and 3.

For metrics which are too complicated, the computer will exhaust its memory capacity before completing the computations. When this occurs, the computer prints out "ERROR TRACE. CALLS IN REVERSE ORDER.", followed by a list of routines being used at that moment, and then followed by "FORMAC ERROR NUMBER '7337' HAS OCCURRED. EXECUTION TERMINATED." The computer then terminates the computation. Other error messages may occur as a result of faulty input. In order to interpret these error messages the reader should refer to pp. 213-222 of the FORMAC manual (IBM 1964).

As an example of successful output from a computation by EINSTEIN, we give below the results for the static, spherically-symmetric metric of equation (1). The input which generated these results was given in § III.B.1 and also in the listing of EINSTEIN in § III.B.2.

```
COVARIANT METRIC TENSOR NON ZERO ELEMENTS  (INPUT)
     1L     1L
      -FMCEXP(ALAM)$
     2L     2L
      -X**2.0$
     3L     3L
      -X**2.0*FMCSIN(Y)**2.0$
     4L     4L
      FMCEXP(ANU)$

CONTRAVARIANT METRIC TENSOR NON ZERO ELEMENTS (INPUT)
     1U     1U
      -FMCEXP(-ALAM)$
     2U     2U
      -X**(-2.0)$
     3U     3U
      -X**(-2.0)*FMCSIN(Y)**(-2.0)$
     4U     4U
      FMCEXP(-ANU)$


THE COVARIANT CHRISTOFFEL SYMBOLS
     1LS  1LS    1L
      -5.0E-1*FMCEXP(ALAM)*FMCDIF(ALAM,(X,1))$
     1LS  2LS    2L
      -X$
     1LS  3LS    3L
      -X*FMCSIN(Y)**2.0$
     1LS  4LS    4L
      5.0E-1*FMCEXP(ANU)*FMCDIF(ANU,(X,1))$
     2LS  2LS    1L
      X$
     2LS  3LS    3L
      -X**2.0*FMCSIN(Y)*FMCCOS(Y)$
     3LS  3LS    1L
      X*FMCSIN(Y)**2.0$
     3LS  3LS    2L
      X**2.0*FMCSIN(Y)*FMCCOS(Y)$
     4LS  4LS    1L
      -5.0E-1*FMCEXP(ANU)*FMCDIF(ANU,(X,1))$


THE MIXED CHRISTOFFEL SYMBOLS
     1LS  1LS    1U
      5.0E-1*FMCDIF(ALAM,(X,1))$
     1LS  2LS    2U
      X**(-1.0)$
     1LS  3LS    3U
      X**(-1.0)$
     1LS  4LS    4U
      5.0E-1*FMCDIF(ANU,(X,1))$
     2LS  2LS    1U
      -X*FMCEXP(-ALAM)$
     2LS  3LS    3U
      FMCSIN(Y)**(-1.0)*FMCCOS(Y)$
     3LS  3LS    1U
      -X*FMCSIN(Y)**2.0*FMCEXP(-ALAM)$
     3LS  3LS    2U
      -FMCSIN(Y)*FMCCOS(Y)$
     4LS  4LS    1U
      5.0E-1*FMCEXP(-ALAM+ANU)*FMCDIF(ANU,(X,1))$
```

```
THE RICCI TENSOR, MIXED COMPONENTS
     1U      1L
      -X**(-1.0)*FMCEXP(-ALAM)*FMCDIF(ALAM,(X,1))-2.5E-1*FMCEXP(-ALAM)*
      FMCDIF(ALAM,(X,1))*FMCDIF(ANU,(X,1))+2.5E-1*FMCEXP(-ALAM)*FMCDIF(
      ANU,(X,1))**2.0+5.0E-1*FMCEXP(-ALAM)*FMCDIF(ANU,(X,2))$
     2U      2L
      X**(-2.0)*FMCEXP(-ALAM)-X**(-2.0)-5.0E-1*X**(-1.0)*FMCEXP(-ALAM)*
      FMCDIF(ALAM,(X,1))+5.0E-1*X**(-1.0)*FMCEXP(-ALAM)*FMCDIF(ANU,(X,1)
      )$
     3U      3L
      X**(-2.0)*FMCEXP(-ALAM)-X**(-2.0)-5.0E-1*X**(-1.0)*FMCEXP(-ALAM)*
      FMCDIF(ALAM,(X,1))+5.0E-1*X**(-1.0)*FMCEXP(-ALAM)*FMCDIF(ANU,(X,1)
      )$
     4U      4L
      X**(-1.0)*FMCEXP(-ALAM)*FMCDIF(ANU,(X,1))-2.5E-1*FMCEXP(-ALAM)*
      FMCDIF(ALAM,(X,1))*FMCDIF(ANU,(X,1))+2.5E-1*FMCEXP(-ALAM)*FMCDIF(
      ANU,(X,1))**2.0+5.0E-1*FMCEXP(-ALAM)*FMCDIF(ANU,(X,2))$
```

```
THE RICCI SCALAR
      2.0*X**(-2.0)*FMCEXP(-ALAM)-2.0*X**(-2.0)-2.0*X**(-1.0)*FMCEXP(-
      ALAM)*FMCDIF(ALAM,(X,1))+2.0*X**(-1.0)*FMCEXP(-ALAM)*FMCDIF(ANU,(X
      ,1))-5.0E-1*FMCEXP(-ALAM)*FMCDIF(ALAM,(X,1))*FMCDIF(ANU,(X,1))+
      5.0E-1*FMCEXP(-ALAM)*FMCDIF(ANU,(X,1))**2.0+FMCEXP(-ALAM)*FMCDIF(
      ANU,(X,2))$
```

```
THE EINSTEIN TENSOR, MIXED COMPONENTS, DIAGONAL ELEMENTS
     1U      1L
      -X**(-2.0)*FMCEXP(-ALAM)+X**(-2.0)-X**(-1.0)*FMCEXP(-ALAM)*FMCDIF(
      ANU,(X,1))$
     2U      2L
      5.0E-1*X**(-1.0)*FMCEXP(-ALAM)*FMCDIF(ALAM,(X,1))-5.0E-1*X**(-1.0)
      *FMCEXP(-ALAM)*FMCDIF(ANU,(X,1))+2.5E-1*FMCEXP(-ALAM)*FMCDIF(ALAM,
      (X,1))*FMCDIF(ANU,(X,1))-2.5E-1*FMCEXP(-ALAM)*FMCDIF(ANU,(X,1))**
      2.0-5.0E-1*FMCEXP(-ALAM)*FMCDIF(ANU,(X,2))$
     3U      3L
      5.0E-1*X**(-1.0)*FMCEXP(-ALAM)*FMCDIF(ALAM,(X,1))-5.0E-1*X**(-1.0)
      *FMCEXP(-ALAM)*FMCDIF(ANU,(X,1))+2.5E-1*FMCEXP(-ALAM)*FMCDIF(ALAM,
      (X,1))*FMCDIF(ANU,(X,1))-2.5E-1*FMCEXP(-ALAM)*FMCDIF(ANU,(X,1))**
      2.0-5.0E-1*FMCEXP(-ALAM)*FMCDIF(ANU,(X,2))$
     4U      4L
      -X**(-2.0)*FMCEXP(-ALAM)+X**(-2.0)+X**(-1.0)*FMCEXP(-ALAM)*FMCDIF(
      ALAM,(X,1))$
```

4. Sample Problems for EINSTEIN

Input and output for one sample problem, the static spherically symmetric metric of equation (1), were given in the last two sections. As a second and much more complicated example, consider the metric for a star in even-parity, nonradial pulsation, as discussed by Thorne and Campolattaro (1967):

$$\begin{aligned} ds^2 = \; & e^{\nu} (1 + H_0 P_\ell) dt^2 + 2 H_1 P_\ell \, dt dr - e^{\lambda} (1 - H_2 P_\ell) dr^2 \\ & - r^2 (1 - K P_\ell) (d\theta^2 + \sin^2\theta d\varphi^2). \end{aligned} \qquad (7)$$

Here ν and λ are functions of r; H_0, H_1, H_2 and K are functions of r and t; and $P_\ell(\cos\theta)$ is the Legendre polynomial of order ℓ.

To prepare input for computing the Ricci and Einstein tensors of the metric (2), we make the following changes to FORMAC notation:

$$\left.\begin{array}{lcl} t & \rightarrow & T \\ r & \rightarrow & X \\ \theta & \rightarrow & Y \\ \varphi & \rightarrow & Z \end{array}\right\} \text{(these are the standard coordinates as accepted by ALBERT).}$$

$$\left.\begin{array}{lcl} \lambda & \rightarrow & ALAM \\ \nu & \rightarrow & ANU \\ H_0 & \rightarrow & H0 \\ H_1 & \rightarrow & H1 \\ H_2 & \rightarrow & H2 \\ K & \rightarrow & AK \\ P_\ell & \rightarrow & AL \end{array}\right\} \text{(these names were chosen arbitrarily, subject only to the rules outlined in § III.B.1.)}$$

Also, we make the decision to let the computer treat $P_\ell(\cos\theta) = AL(Y)$ as an unspecified function. Finally, because we want the Ricci and Einstein tensors only to first order (linear) in the perturbation from equilibrium, we attach the expansion parameter EP**2 to the metric perturbations, AK, H0, H1, and H2, wherever they appear. The resultant input and output are as follows:

INPUT:

```
 ATOMIC X,Y,Z,T,EP,AL,ANU,ALAM,AK,H0,H1,H2
 DEPEND (ANU,ALAM/X), (AL/Y)
 DEPEND (H0,H1,H2,AK/X,T)


  EPTERM = 3


LET G1(1,1) = -(-AL*EP**2*H2+1.)*FMCEXP(ALAM)
LET G1(1,4) =  AL*EP**2*H1
LET G1(4,1) =  G1(1,4)
LET G1(2,2) = -(1.-AK*AL*EP**2)*X**2
LET G1(3,3) = -(1.-AK*AL*EP**2)*X**2*FMCSIN(Y)**2
LET G1(4,4) =  (AL*EP**2*H0+1.)*FMCEXP(ANU)
LET G2(1,1) = -(AL*EP**2*H2+1.)*FMCEXP(-ALAM)
LET G2(1,4) =  AL*EP**2*H1*FMCEXP(-ALAM-ANU)
LET G2(4,1) =  G2(1,4)
LET G2(2,2) = -(AK*AL*EP**2+1.)*X**(-2)
LET G2(3,3) = -(AK*AL*EP**2+1.)*(X*FMCSIN(Y))**(-2)
LET G2(4,4) =  (-AL*EP**2*H0+1.)*FMCEXP(-ANU)
```

OUTPUT:

See following pages

```
  COVARIANT METRIC TENSOR NON ZERO ELEMENTS  (INPUT)
      1L     1L
       -(-AL*EP**2.0*H2+1.0)*FMCEXP(ALAM)$
      1L     4L
       AL*EP**2.0*H1$
      2L     2L
       -(-AK*AL*EP**2.0+1.0)*X**2.0$
      3L     3L
       -(-AK*AL*EP**2.0+1.0)*X**2.0*FMCSIN(Y)**2.0$
      4L     1L
       AL*EP**2.0*H1$
      4L     4L
       (AL*EP**2.0*H0+1.0)*FMCEXP(ANU)$

  CONTRAVARIANT METRIC TENSOR NON ZERO ELEMENTS (INPUT)
      1U     1U
       -(AL*EP**2.0*H2+1.0)*FMCEXP(-ALAM)$
      1U     4U
       AL*EP**2.0*H1*FMCEXP(-ALAM-ANU)$
      2U     2U
       -(AK*AL*EP**2.0+1.0)*X**(-2.0)$
      3U     3U
       -(AK*AL*EP**2.0+1.0)*X**(-2.0)*FMCSIN(Y)**(-2.0)$
      4U     1U
       AL*EP**2.0*H1*FMCEXP(-ALAM-ANU)$
      4U     4U
       (-AL*EP**2.0*H0+1.0)*FMCEXP(-ANU)$



THE COVARIANT CHRISTOFFEL SYMBOLS
    1LS  1LS    1L
     5.0E-1*AL*EP**2.0*H2*FMCEXP(ALAM)*FMCDIF(ALAM,(X,1))+5.0E-1*AL*EP
     **2.0*FMCEXP(ALAM)*FMCDIF(H2,(X,1))-5.0E-1*FMCEXP(ALAM)*FMCDIF(
     ALAM,(X,1))$
    1LS  1LS    2L
     -5.0E-1*EP**2.0*H2*FMCEXP(ALAM)*FMCDIF(AL,(Y,1))$
    1LS  1LS    4L
     -5.0E-1*AL*EP**2.0*FMCEXP(ALAM)*FMCDIF(H2,(T,1))+AL*EP**2.0*FMCDIF
     (H1,(X,1))$
    1LS  2LS    1L
     5.0E-1*EP**2.0*H2*FMCEXP(ALAM)*FMCDIF(AL,(Y,1))$
    1LS  2LS    2L
     AK*AL*EP**2.0*X+5.0E-1*AL*EP**2.0*X**2.0*FMCDIF(AK,(X,1))-X$
    1LS  2LS    4L
     5.0E-1*EP**2.0*H1*FMCDIF(AL,(Y,1))$
    1LS  3LS    3L
     AK*AL*EP**2.0*X*FMCSIN(Y)**2.0+5.0E-1*AL*EP**2.0*X**2.0*FMCSIN(Y)
     **2.0*FMCDIF(AK,(X,1))-X*FMCSIN(Y)**2.0$
    1LS  4LS    1L
     5.0E-1*AL*EP**2.0*FMCEXP(ALAM)*FMCDIF(H2,(T,1))$
    1LS  4LS    2L
     -5.0E-1*EP**2.0*H1*FMCDIF(AL,(Y,1))$
    1LS   4LS    4L
     5.0E-1*AL*EP**2.0*H0*FMCEXP(ANU)*FMCDIF(ANU,(X,1))+5.0E-1*AL*EP**
     2.0*FMCEXP(ANU)*FMCDIF(H0,(X,1))+5.0E-1*FMCEXP(ANU)*FMCDIF(ANU,(X,
     1))$
    2LS   2LS    1L
     -AK*AL*EP**2.0*X-5.0E-1*AL*EP**2.0*X**2.0*FMCDIF(AK,(X,1))+X$
    2LS   2LS    2L
     5.0E-1*AK*EP**2.0*X**2.0*FMCDIF(AL,(Y,1))$
    2LS   2LS    4L
     -5.0E-1*AL*EP**2.0*X**2.0*FMCDIF(AK,(T,1))$
    2LS   3LS    3L
     AK*AL*EP**2.0*X**2.0*FMCSIN(Y)*FMCCOS(Y)+5.0E-1*AK*EP**2.0*X**2.0*
     FMCSIN(Y)**2.0*FMCDIF(AL,(Y,1))-X**2.0*FMCSIN(Y)*FMCCOS(Y)$
    2LS   4LS    1L
     5.0E-1*EP**2.0*H1*FMCDIF(AL,(Y,1))$
    2LS   4LS    2L
     5.0E-1*AL*EP**2.0*X**2.0*FMCDIF(AK,(T,1))$
    2LS   4LS    4L
     5.0E-1*EP**2.0*H0*FMCEXP(ANU)*FMCDIF(AL,(Y,1))$
    3LS   3LS    1L
     -AK*AL*EP**2.0*X*FMCSIN(Y)**2.0-5.0E-1*AL*EP**2.0*X**2.0*FMCSIN(Y)
     **2.0*FMCDIF(AK,(X,1))+X*FMCSIN(Y)**2.0$
    3LS   3LS    2L
     -AK*AL*EP**2.0*X**2.0*FMCSIN(Y)*FMCCOS(Y)-5.0E-1*AK*EP**2.0*X**2.0
     *FMCSIN(Y)**2.0*FMCDIF(AL,(Y,1))+X**2.0*FMCSIN(Y)*FMCCOS(Y)$
    3LS   3LS    4L
     -5.0E-1*AL*EP**2.0*X**2.0*FMCSIN(Y)**2.0*FMCDIF(AK,(T,1))$
    3LS   4LS    3L
     5.0E-1*AL*EP**2.0*X**2.0*FMCSIN(Y)**2.0*FMCDIF(AK,(T,1))$
    4LS   4LS    1L
     -5.0E-1*AL*EP**2.0*H0*FMCEXP(ANU)*FMCDIF(ANU,(X,1))-5.0E-1*AL*EP**
     2.0*FMCEXP(ANU)*FMCDIF(H0,(X,1))+AL*EP**2.0*FMCDIF(H1,(T,1))-
     5.0E-1*FMCEXP(ANU)*FMCDIF(ANU,(X,1))$
    4LS   4LS    2L
     -5.0E-1*EP**2.0*H0*FMCEXP(ANU)*FMCDIF(AL,(Y,1))$
    4LS   4LS    4L
     5.0E-1*AL*EP**2.0*FMCEXP(ANU)*FMCDIF(H0,(T,1))$



THE MIXED CHRISTOFFEL SYMBOLS
    1LS   1LS    1U
     -5.0E-1*AL*EP**2.0*FMCDIF(H2,(X,1))+5.0E-1*FMCDIF(ALAM,(X,1))$
    1LS   1LS    2U
     5.0E-1*EP**2.0*H2*X**(-2.0)*FMCEXP(ALAM)*FMCDIF(AL,(Y,1))$
    1LS   1LS    4U
     -5.0E-1*AL*EP**2.0*H1*FMCEXP(-ANU)*FMCDIF(ALAM,(X,1))-5.0E-1*AL*EP
     **2.0*FMCEXP(ALAM-ANU)*FMCDIF(H2,(T,1))+AL*EP**2.0*FMCEXP(-ANU)*
     FMCDIF(H1,(X,1))$
    1LS   2LS    1U
     -5.0E-1*EP**2.0*H2*FMCDIF(AL,(Y,1))$
    1LS   2LS    2U
     -5.0E-1*AL*EP**2.0*FMCDIF(AK,(X,1))+X**(-1.0)$
```

```
1LS  2LS   4U
 5.0E-1*EP**2.0*H1*FMCEXP(-ANU)*FMCDIF(AL,(Y,1))$
1LS  3LS   3U
 -5.0E-1*AL*EP**2.0*FMCDIF(AK,(X,1))+X**(-1.0)$
1LS  4LS   1U
 5.0E-1*AL*EP**2.0*H1*FMCEXP(-ALAM)*FMCDIF(ANU,(X,1))-5.0E-1*AL*EP
 **2.0*FMCDIF(H2,(T,1))$
1LS  4LS   2U
 5.0E-1*EP**2.0*H1*X**(-2.0)*FMCDIF(AL,(Y,1))$
1LS  4LS   4U
 5.0E-1*AL*EP**2.0*FMCDIF(HO,(X,1))+5.0E-1*FMCDIF(ANU,(X,1))$
2LS  2LS   1U
 AK*AL*EP**2.0*X*FMCEXP(-ALAM)-AL*EP**2.0*H2*X*FMCEXP(-ALAM)+
 5.0E-1*AL*EP**2.0*X**2.0*FMCEXP(-ALAM)*FMCDIF(AK,(X,1))-X*FMCEXP(-
 ALAM)$
2LS  2LS   2U
 -5.0E-1*AK*EP**2.0*FMCDIF(AL,(Y,1))$
2LS  2LS   4U
 AL*EP**2.0*H1*X*FMCEXP(-ALAM-ANU)-5.0E-1*AL*EP**2.0*X**2.0*FMCEXP(
 -ANU)*FMCDIF(AK,(T,1))$
2LS  3LS   3U
 -5.0E-1*AK*EP**2.0*FMCDIF(AL,(Y,1))+FMCSIN(Y)**(-1.0)*FMCCOS(Y)$
2LS  4LS   1U
 -5.0E-1*EP**2.0*H1*FMCEXP(-ALAM)*FMCDIF(AL,(Y,1))$
2LS  4LS   2U
 -5.0E-1*AL*EP**2.0*FMCDIF(AK,(T,1))$
2LS  4LS   4U
 5.0E-1*EP**2.0*HO*FMCDIF(AL,(Y,1))$
3LS  3LS   1U
 AK*AL*EP**2.0*X*FMCSIN(Y)**2.0*FMCEXP(-ALAM)-AL*EP**2.0*H2*X*
 FMCSIN(Y)**2.0*FMCEXP(-ALAM)+5.0E-1*AL*EP**2.0*X**2.0*FMCSIN(Y)**
 2.0*FMCEXP(-ALAM)*FMCDIF(AK,(X,1))-X*FMCSIN(Y)**2.0*FMCEXP(-ALAM)$
3LS  3LS   2U
 5.0E-1*AK*EP**2.0*FMCSIN(Y)**2.0*FMCDIF(AL,(Y,1))-FMCSIN(Y)*FMCCOS
 (Y)$
3LS  3LS   4U
 AL*EP**2.0*H1*X*FMCSIN(Y)**2.0*FMCEXP(-ALAM-ANU)-5.0E-1*AL*EP**2.0
 *X**2.0*FMCSIN(Y)**2.0*FMCEXP(-ANU)*FMCDIF(AK,(T,1))$
3LS  4LS   3U
 -5.0E-1*AL*EP**2.0*FMCDIF(AK,(T,1))$
4LS  4LS   1U
 5.0E-1*AL*EP**2.0*HO*FMCEXP(-ALAM+ANU)*FMCDIF(ANU,(X,1))+
 5.0E-1*AL*EP**2.0*H2*FMCEXP(-ALAM+ANU)*FMCDIF(ANU,(X,1))-AL*EP**
 2.0*FMCEXP(-ALAM)*FMCDIF(H1,(T,1))+5.0E-1*AL*EP**2.0*FMCEXP(-ALAM+
 ANU)*FMCDIF(HO,(X,1))+5.0E-1*FMCEXP(-ALAM+ANU)*FMCDIF(ANU,(X,1))$
4LS  4LS   2U
 5.0E-1*EP**2.0*HO*X**(-2.0)*FMCEXP(ANU)*FMCDIF(AL,(Y,1))$
4LS  4LS   4U
 -5.0E-1*AL*EP**2.0*H1*FMCEXP(-ALAM)*FMCDIF(ANU,(X,1))+5.0E-1*AL*EP
 **2.0*FMCDIF(HO,(T,1))$
```

```
THE RICCI TENSOR, MIXED COMPONENTS
    1U      1L
     -AL*EP**2.0*H2*X**(-1.0)*FMCEXP(-ALAM)*FMCDIF(ALAM,(X,1))-
     2.5E-1*AL*EP**2.0*H2*FMCEXP(-ALAM)*FMCDIF(ALAM,(X,1))*FMCDIF(ANU,(
     X,1))+2.5E-1*AL*EP**2.0*H2*FMCEXP(-ALAM)*FMCDIF(ANU,(X,1))**2.0+
     5.0E-1*AL*EP**2.0*H2*FMCEXP(-ALAM)*FMCDIF(ANU,(X,2))-2.0*AL*EP**
     2.0*X**(-1.0)*FMCEXP(-ALAM)*FMCDIF(AK,(X,1))+AL*EP**2.0*X**(-1.0)*
     FMCEXP(-ALAM)*FMCDIF(H2,(X,1))+5.0E-1*AL*EP**2.0*FMCEXP(-ALAM)*
     FMCDIF(AK,(X,1))*FMCDIF(ALAM,(X,1))-AL*EP**2.0*FMCEXP(-ALAM)*
     FMCDIF(AK,(X,2))-2.5E-1*AL*EP**2.0*FMCEXP(-ALAM)*FMCDIF(ALAM,(X,1)
     )*FMCDIF(H0,(X,1))+5.0E-1*AL*EP**2.0*FMCEXP(-ALAM)*FMCDIF(ANU,(X,1
     ))*FMCDIF(H0,(X,1))+2.5E-1*AL*EP**2.0*FMCEXP(-ALAM)*FMCDIF(ANU,(X,
     1))*FMCDIF(H2,(X,1))+5.0E-1*AL*EP**2.0*FMCEXP(-ALAM)*FMCDIF(H0,(X,
     2))+5.0E-1*AL*EP**2.0*FMCEXP(-ALAM-ANU)*FMCDIF(ALAM,(X,1))*FMCDIF(
     H1,(T,1))-AL*EP**2.0*FMCEXP(-ALAM-ANU)*FMCDIF(H1,(T,1),(X,1))+
     5.0E-1*AL*EP**2.0*FMCEXP(-ANU)*FMCDIF(H2,(T,2))-5.0E-1*EP**2.0*H2*
     X**(-2.0)*FMCSIN(Y)**(-1.0)*FMCCOS(Y)*FMCDIF(AL,(Y,1))-5.0E-1*EP**
     2.0*H2*X**(-2.0)*FMCDIF(AL,(Y,2))-X**(-1.0)*FMCEXP(-ALAM)*FMCDIF(
     ALAM,(X,1))-2.5E-1*FMCEXP(-ALAM)*FMCDIF(ALAM,(X,1))*FMCDIF(ANU,(X,
     1))+2.5E-1*FMCEXP(-ALAM)*FMCDIF(ANU,(X,1))**2.0+5.0E-1*FMCEXP(-
     ALAM)*FMCDIF(ANU,(X,2))$
    2U      1L
     -5.0E-1*EP**2.0*H0*X**(-3.0)*FMCDIF(AL,(Y,1))+2.5E-1*EP**2.0*H0*X
     **(-2.0)*FMCDIF(AL,(Y,1))*FMCDIF(ANU,(X,1))+5.0E-1*EP**2.0*H2*X**(
     -3.0)*FMCDIF(AL,(Y,1))+2.5E-1*EP**2.0*H2*X**(-2.0)*FMCDIF(AL,(Y,1)
     )*FMCDIF(ANU,(X,1))-5.0E-1*EP**2.0*X**(-2.0)*FMCEXP(-ANU)*FMCDIF(
     AL,(Y,1))*FMCDIF(H1,(T,1))-5.0E-1*EP**2.0*X**(-2.0)*FMCDIF(AK,(X,1
     ))*FMCDIF(AL,(Y,1))+5.0E-1*EP**2.0*X**(-2.0)*FMCDIF(AL,(Y,1))*
     FMCDIF(H0,(X,1))$
    4U      1L
     AL*EP**2.0*H1*X**(-1.0)*FMCEXP(-ALAM-ANU)*FMCDIF(ALAM,(X,1))+AL*EP
     **2.0*H1*X**(-1.0)*FMCEXP(-ALAM-ANU)*FMCDIF(ANU,(X,1))+AL*EP**2.0*
     X**(-1.0)*FMCEXP(-ANU)*FMCDIF(AK,(T,1))-AL*EP**2.0*X**(-1.0)*
     FMCEXP(-ANU)*FMCDIF(H2,(T,1))-5.0E-1*AL*EP**2.0*FMCEXP(-ANU)*
     FMCDIF(AK,(T,1))*FMCDIF(ANU,(X,1))+AL*EP**2.0*FMCEXP(-ANU)*FMCDIF(
     AK,(T,1),(X,1))+5.0E-1*EP**2.0*H1*X**(-2.0)*FMCSIN(Y)**(-1.0)*
     FMCCOS(Y)*FMCEXP(-ANU)*FMCDIF(AL,(Y,1))+5.0E-1*EP**2.0*H1*X**(-2.0
     )*FMCEXP(-ANU)*FMCDIF(AL,(Y,2))$
    1U      2L
     -5.0E-1*EP**2.0*H0*X**(-1.0)*FMCEXP(-ALAM)*FMCDIF(AL,(Y,1))+
     2.5E-1*EP**2.0*H0*FMCEXP(-ALAM)*FMCDIF(AL,(Y,1))*FMCDIF(ANU,(X,1))
     +5.0E-1*EP**2.0*H2*X**(-1.0)*FMCEXP(-ALAM)*FMCDIF(AL,(Y,1))+
     2.5E-1*EP**2.0*H2*FMCEXP(-ALAM)*FMCDIF(AL,(Y,1))*FMCDIF(ANU,(X,1))
     -5.0E-1*EP**2.0*FMCEXP(-ALAM)*FMCDIF(AK,(X,1))*FMCDIF(AL,(Y,1))+
     5.0E-1*EP**2.0*FMCEXP(-ALAM)*FMCDIF(AL,(Y,1))*FMCDIF(H0,(X,1))-
     5.0E-1*EP**2.0*FMCEXP(-ALAM-ANU)*FMCDIF(AL,(Y,1))*FMCDIF(H1,(T,1))
     $
```

THE RICCI TENSOR (contimued)

```
2U     2L
 -AK*AL*EP**2.0*X**(-2.0)-5.0E-1*AK*EP**2.0*X**(-2.0)*FMCSIN(Y)**(
 -1.0)*FMCCOS(Y)*FMCDIF(AL,(Y,1))-5.0E-1*AK*EP**2.0*X**(-2.0)*
 FMCDIF(AL,(Y,2))+AL*EP**2.0*H2*X**(-2.0)*FMCEXP(-ALAM)-5.0E-1*AL*
 EP**2.0*H2*X**(-1.0)*FMCEXP(-ALAM)*FMCDIF(ALAM,(X,1))+5.0E-1*AL*EP
 **2.0*H2*X**(-1.0)*FMCEXP(-ALAM)*FMCDIF(ANU,(X,1))-2.0*AL*EP**2.0*
 X**(-1.0)*FMCEXP(-ALAM)*FMCDIF(AK,(X,1))+5.0E-1*AL*EP**2.0*X**(
 -1.0)*FMCEXP(-ALAM)*FMCDIF(H0,(X,1))+5.0E-1*AL*EP**2.0*X**(-1.0)*
 FMCEXP(-ALAM)*FMCDIF(H2,(X,1))-AL*EP**2.0*X**(-1.0)*FMCEXP(-ALAM-
 ANU)*FMCDIF(H1,(T,1))+2.5E-1*AL*EP**2.0*FMCEXP(-ALAM)*FMCDIF(AK,(X
 ,1))*FMCDIF(ALAM,(X,1))-2.5E-1*AL*EP**2.0*FMCEXP(-ALAM)*FMCDIF(AK,
 (X,1))*FMCDIF(ANU,(X,1))-5.0E-1*AL*EP**2.0*FMCEXP(-ALAM)*FMCDIF(AK
 ,(X,2))+5.0E-1*AL*EP**2.0*FMCEXP(-ANU)*FMCDIF(AK,(T,2))+5.0E-1*EP
 **2.0*H0*X**(-2.0)*FMCDIF(AL,(Y,2))-5.0E-1*EP**2.0*H2*X**(-2.0)*
 FMCDIF(AL,(Y,2))+X**(-2.0)*FMCEXP(-ALAM)-X**(-2.0)-5.0E-1*X**(-1.0
 )*FMCEXP(-ALAM)*FMCDIF(ALAM,(X,1))+5.0E-1*X**(-1.0)*FMCEXP(-ALAM)*
 FMCDIF(ANU,(X,1))$
```

```
4U     2L
 2.5E-1*EP**2.0*H1*FMCEXP(-ALAM-ANU)*FMCDIF(AL,(Y,1))*FMCDIF(ALAM,(
 X,1))-2.5E-1*EP**2.0*H1*FMCEXP(-ALAM-ANU)*FMCDIF(AL,(Y,1))*FMCDIF(
 ANU,(X,1))-5.0E-1*EP**2.0*FMCEXP(-ALAM-ANU)*FMCDIF(AL,(Y,1))*
 FMCDIF(H1,(X,1))+5.0E-1*EP**2.0*FMCEXP(-ANU)*FMCDIF(AK,(T,1))*
 FMCDIF(AL,(Y,1))+5.0E-1*EP**2.0*FMCEXP(-ANU)*FMCDIF(AL,(Y,1))*
 FMCDIF(H2,(T,1))$
3U     3L
 -AK*AL*EP**2.0*X**(-2.0)-5.0E-1*AK*EP**2.0*X**(-2.0)*FMCSIN(Y)**(
 -1.0)*FMCCOS(Y)*FMCDIF(AL,(Y,1))-5.0E-1*AK*EP**2.0*X**(-2.0)*
 FMCDIF(AL,(Y,2))+AL*EP**2.0*H2*X**(-2.0)*FMCEXP(-ALAM)-5.0E-1*AL*
 EP**2.0*H2*X**(-1.0)*FMCEXP(-ALAM)*FMCDIF(ALAM,(X,1))+5.0E-1*AL*EP
 **2.0*H2*X**(-1.0)*FMCEXP(-ALAM)*FMCDIF(ANU,(X,1))-2.0*AL*EP**2.0*
 X**(-1.0)*FMCEXP(-ALAM)*FMCDIF(AK,(X,1))+5.0E-1*AL*EP**2.0*X**(
 -1.0)*FMCEXP(-ALAM)*FMCDIF(H0,(X,1))+5.0E-1*AL*EP**2.0*X**(-1.0)*
 FMCEXP(-ALAM)*FMCDIF(H2,(X,1))-AL*EP**2.0*X**(-1.0)*FMCEXP(-ALAM-
 ANU)*FMCDIF(H1,(T,1))+2.5E-1*AL*EP**2.0*FMCEXP(-ALAM)*FMCDIF(AK,(X
 ,1))*FMCDIF(ALAM,(X,1))-2.5E-1*AL*EP**2.0*FMCEXP(-ALAM)*FMCDIF(AK,
 (X,1))*FMCDIF(ANU,(X,1))-5.0E-1*AL*EP**2.0*FMCEXP(-ALAM)*FMCDIF(AK
 ,(X,2))+5.0E-1*AL*EP**2.0*FMCEXP(-ANU)*FMCDIF(AK,(T,2))+5.0E-1*EP
 **2.0*H0*X**(-2.0)*FMCSIN(Y)**(-1.0)*FMCCOS(Y)*FMCDIF(AL,(Y,1))-
 5.0E-1*EP**2.0*H2*X**(-2.0)*FMCSIN(Y)**(-1.0)*FMCCOS(Y)*FMCDIF(AL,
 (Y,1))+X**(-2.0)*FMCEXP(-ALAM)-X**(-2.0)-5.0E-1*X**(-1.0)*FMCEXP(-
 ALAM)*FMCDIF(ALAM,(X,1))+5.0E-1*X**(-1.0)*FMCEXP(-ALAM)*FMCDIF(ANU
 ,(X,1))$
1U     4L
 -AL*EP**2.0*X**(-1.0)*FMCEXP(-ALAM)*FMCDIF(AK,(T,1))+AL*EP**2.0*X
 **(-1.0)*FMCEXP(-ALAM)*FMCDIF(H2,(T,1))+5.0E-1*AL*EP**2.0*FMCEXP(-
 ALAM)*FMCDIF(AK,(T,1))*FMCDIF(ANU,(X,1))-AL*EP**2.0*FMCEXP(-ALAM)*
 FMCDIF(AK,(T,1),(X,1))-5.0E-1*EP**2.0*H1*X**(-2.0)*FMCSIN(Y)**(
 -1.0)*FMCCOS(Y)*FMCEXP(-ALAM)*FMCDIF(AL,(Y,1))-5.0E-1*EP**2.0*H1*X
 **(-2.0)*FMCEXP(-ALAM)*FMCDIF(AL,(Y,2))$
2U     4L
 -2.5E-1*EP**2.0*H1*X**(-2.0)*FMCEXP(-ALAM)*FMCDIF(AL,(Y,1))*FMCDIF
 (ALAM,(X,1))+2.5E-1*EP**2.0*H1*X**(-2.0)*FMCEXP(-ALAM)*FMCDIF(AL,(
 Y,1))*FMCDIF(ANU,(X,1))+5.0E-1*EP**2.0*X**(-2.0)*FMCEXP(-ALAM)*
 FMCDIF(AL,(Y,1))*FMCDIF(H1,(X,1))-5.0E-1*EP**2.0*X**(-2.0)*FMCDIF(
 AK,(T,1))*FMCDIF(AL,(Y,1))-5.0E-1*EP**2.0*X**(-2.0)*FMCDIF(AL,(Y,1
 ))*FMCDIF(H2,(T,1))$
4U     4L
 AL*EP**2.0*H2*X**(-1.0)*FMCEXP(-ALAM)*FMCDIF(ANU,(X,1))-2.5E-1*AL*
 EP**2.0*H2*FMCEXP(-ALAM)*FMCDIF(ALAM,(X,1))*FMCDIF(ANU,(X,1))+
 2.5E-1*AL*EP**2.0*H2*FMCEXP(-ALAM)*FMCDIF(ANU,(X,1))**2.0+
 5.0E-1*AL*EP**2.0*H2*FMCEXP(-ALAM)*FMCDIF(ANU,(X,2))+AL*EP**2.0*X
 **(-1.0)*FMCEXP(-ALAM)*FMCDIF(H0,(X,1))-2.0*AL*EP**2.0*X**(-1.0)*
 FMCEXP(-ALAM-ANU)*FMCDIF(H1,(T,1))-5.0E-1*AL*EP**2.0*FMCEXP(-ALAM)
 *FMCDIF(AK,(X,1))*FMCDIF(ANU,(X,1))-2.5E-1*AL*EP**2.0*FMCEXP(-ALAM
 )*FMCDIF(ALAM,(X,1))*FMCDIF(H0,(X,1))+5.0E-1*AL*EP**2.0*FMCEXP(-
 ALAM)*FMCDIF(ANU,(X,1))*FMCDIF(H0,(X,1))+2.5E-1*AL*EP**2.0*FMCEXP(
 -ALAM)*FMCDIF(ANU,(X,1))*FMCDIF(H2,(X,1))+5.0E-1*AL*EP**2.0*FMCEXP
 (-ALAM)*FMCDIF(H0,(X,2))+5.0E-1*AL*EP**2.0*FMCEXP(-ALAM-ANU)*
 FMCDIF(ALAM,(X,1))*FMCDIF(H1,(T,1))-AL*EP**2.0*FMCEXP(-ALAM-ANU)*
 FMCDIF(H1,(T,1),(X,1))+AL*EP**2.0*FMCEXP(-ANU)*FMCDIF(AK,(T,2))+
 5.0E-1*AL*EP**2.0*FMCEXP(-ANU)*FMCDIF(H2,(T,2))+5.0E-1*EP**2.0*H0*
 X**(-2.0)*FMCSIN(Y)**(-1.0)*FMCCOS(Y)*FMCDIF(AL,(Y,1))+5.0E-1*EP**
 2.0*H0*X**(-2.0)*FMCDIF(AL,(Y,2))+X**(-1.0)*FMCEXP(-ALAM)*FMCDIF(
 ANU,(X,1))-2.5E-1*FMCEXP(-ALAM)*FMCDIF(ALAM,(X,1))*FMCDIF(ANU,(X,1
 ))+2.5E-1*FMCEXP(-ALAM)*FMCDIF(ANU,(X,1))**2.0+5.0E-1*FMCEXP(-ALAM
 )*FMCDIF(ANU,(X,2))$
```

```
THE RICCI SCALAR
     -2.0*AK*AL*EP**2.0*X**(-2.0)-AK*EP**2.0*X**(-2.0)*FMCSIN(Y)**(-1.0
     )*FMCCOS(Y)*FMCDIF(AL,(Y,1))-AK*EP**2.0*X**(-2.0)*FMCDIF(AL,(Y,2))
     +2.0*AL*EP**2.0*H2*X**(-2.0)*FMCEXP(-ALAM)-2.0*AL*EP**2.0*H2*X**(
     -1.0)*FMCEXP(-ALAM)*FMCDIF(ALAM,(X,1))+2.0*AL*EP**2.0*H2*X**(-1.0)
     *FMCEXP(-ALAM)*FMCDIF(ANU,(X,1))-5.0E-1*AL*EP**2.0*H2*FMCEXP(-ALAM
     )*FMCDIF(ALAM,(X,1))*FMCDIF(ANU,(X,1))+5.0E-1*AL*EP**2.0*H2*FMCEXP
     (-ALAM)*FMCDIF(ANU,(X,1))**2.0+AL*EP**2.0*H2*FMCEXP(-ALAM)*FMCDIF(
     ANU,(X,2))-6.0*AL*EP**2.0*X**(-1.0)*FMCEXP(-ALAM)*FMCDIF(AK,(X,1))
     +2.0*AL*EP**2.0*X**(-1.0)*FMCEXP(-ALAM)*FMCDIF(H0,(X,1))+2.0*AL*EP
     **2.0*X**(-1.0)*FMCEXP(-ALAM)*FMCDIF(H2,(X,1))-4.0*AL*EP**2.0*X**(
     -1.0)*FMCEXP(-ALAM-ANU)*FMCDIF(H1,(T,1))+AL*EP**2.0*FMCEXP(-ALAM)*
     FMCDIF(AK,(X,1))*FMCDIF(ALAM,(X,1))-AL*EP**2.0*FMCEXP(-ALAM)*
     FMCDIF(AK,(X,1))*FMCDIF(ANU,(X,1))-2.0*AL*EP**2.0*FMCEXP(-ALAM)*
     FMCDIF(AK,(X,2))-5.0E-1*AL*EP**2.0*FMCEXP(-ALAM)*FMCDIF(ALAM,(X,1)
     )*FMCDIF(H0,(X,1))+AL*EP**2.0*FMCEXP(-ALAM)*FMCDIF(ANU,(X,1))*
     FMCDIF(H0,(X,1))+5.0E-1*AL*EP**2.0*FMCEXP(-ALAM)*FMCDIF(ANU,(X,1))
     *FMCDIF(H2,(X,1))+AL*EP**2.0*FMCEXP(-ALAM)*FMCDIF(H0,(X,2))+AL*EP
     **2.0*FMCEXP(-ALAM-ANU)*FMCDIF(ALAM,(X,1))*FMCDIF(H1,(T,1))-2.0*AL
     *EP**2.0*FMCEXP(-ALAM-ANU)*FMCDIF(H1,(T,1),(X,1))+2.0*AL*EP**2.0*
     FMCEXP(-ANU)*FMCDIF(AK,(T,2))+AL*EP**2.0*FMCEXP(-ANU)*FMCDIF(H2,(T
     ,2))+EP**2.0*H0*X**(-2.0)*FMCSIN(Y)**(-1.0)*FMCCOS(Y)*FMCDIF(AL,(Y
     ,1))+EP**2.0*H0*X**(-2.0)*FMCDIF(AL,(Y,2))-EP**2.0*H2*X**(-2.0)*
     FMCSIN(Y)**(-1.0)*FMCCOS(Y)*FMCDIF(AL,(Y,1))-EP**2.0*H2*X**(-2.0)*
     FMCDIF(AL,(Y,2))+2.0*X**(-2.0)*FMCEXP(-ALAM)-2.0*X**(-2.0)-2.0*X**
     (-1.0)*FMCEXP(-ALAM)*FMCDIF(ALAM,(X,1))+2.0*X**(-1.0)*FMCEXP(-ALAM
     )*FMCDIF(ANU,(X,1))-5.0E-1*FMCEXP(-ALAM)*FMCDIF(ALAM,(X,1))*FMCDIF
     (ANU,(X,1))+5.0E-1*FMCEXP(-ALAM)*FMCDIF(ANU,(X,1))**2.0+FMCEXP(-
     ALAM)*FMCDIF(ANU,(X,2))$
```

```
THE EINSTEIN TENSOR, MIXED COMPONENTS, DIAGONAL ELEMENTS
   1U     1L
    AK*AL*EP**2.0*X**(-2.0)+5.0E-1*AK*EP**2.0*X**(-2.0)*FMCSIN(Y)**(
    -1.0)*FMCCOS(Y)*FMCDIF(AL,(Y,1))+5.0E-1*AK*EP**2.0*X**(-2.0)*
    FMCDIF(AL,(Y,2))-AL*EP**2.0*H2*X**(-2.0)*FMCEXP(-ALAM)-AL*EP**2.0*
    H2*X**(-1.0)*FMCEXP(-ALAM)*FMCDIF(ANU,(X,1))+AL*EP**2.0*X**(-1.0)*
    FMCEXP(-ALAM)*FMCDIF(AK,(X,1))-AL*EP**2.0*X**(-1.0)*FMCEXP(-ALAM)*
    FMCDIF(HO,(X,1))+2.0*AL*EP**2.0*X**(-1.0)*FMCEXP(-ALAM-ANU)*FMCDIF
    (H1,(T,1))+5.0E-1*AL*EP**2.0*FMCEXP(-ALAM)*FMCDIF(AK,(X,1))*FMCDIF
    (ANU,(X,1))-AL*EP**2.0*FMCEXP(-ANU)*FMCDIF(AK,(T,2))-5.0E-1*EP**
    2.0*HO*X**(-2.0)*FMCSIN(Y)**(-1.0)*FMCCOS(Y)*FMCDIF(AL,(Y,1))-
    5.0E-1*EP**2.0*HO*X**(-2.0)*FMCDIF(AL,(Y,2))-X**(-2.0)*FMCEXP(-
    ALAM)+X**(-2.0)-X**(-1.0)*FMCEXP(-ALAM)*FMCDIF(ANU,(X,1))$
   2U     2L
    5.0E-1*AL*EP**2.0*H2*X**(-1.0)*FMCEXP(-ALAM)*FMCDIF(ALAM,(X,1))-
    5.0E-1*AL*EP**2.0*H2*X**(-1.0)*FMCEXP(-ALAM)*FMCDIF(ANU,(X,1))+
    2.5E-1*AL*EP**2.0*H2*FMCEXP(-ALAM)*FMCDIF(ALAM,(X,1))*FMCDIF(ANU,(
    X,1))-2.5E-1*AL*EP**2.0*H2*FMCEXP(-ALAM)*FMCDIF(ANU,(X,1))**2.0-
    5.0E-1*AL*EP**2.0*H2*FMCEXP(-ALAM)*FMCDIF(ANU,(X,2))+AL*EP**2.0*X
    **(-1.0)*FMCEXP(-ALAM)*FMCDIF(AK,(X,1))-5.0E-1*AL*EP**2.0*X**(-1.0
    )*FMCEXP(-ALAM)*FMCDIF(HO,(X,1))-5.0E-1*AL*EP**2.0*X**(-1.0)*
    FMCEXP(-ALAM)*FMCDIF(H2,(X,1))+AL*EP**2.0*X**(-1.0)*FMCEXP(-ALAM-
    ANU)*FMCDIF(H1,(T,1))-2.5E-1*AL*EP**2.0*FMCEXP(-ALAM)*FMCDIF(AK,(X
    ,1))*FMCDIF(ALAM,(X,1))+2.5E-1*AL*EP**2.0*FMCEXP(-ALAM)*FMCDIF(AK,
    (X,1))*FMCDIF(ANU,(X,1))+5.0E-1*AL*EP**2.0*FMCEXP(-ALAM)*FMCDIF(AK
    ,(X,2))+2.5E-1*AL*EP**2.0*FMCEXP(-ALAM)*FMCDIF(ALAM,(X,1))*FMCDIF(
    HO,(X,1))-5.0E-1*AL*EP**2.0*FMCEXP(-ALAM)*FMCDIF(ANU,(X,1))*FMCDIF
    (HO,(X,1))-2.5E-1*AL*EP**2.0*FMCEXP(-ALAM)*FMCDIF(ANU,(X,1))*
    FMCDIF(H2,(X,1))-5.0E-1*AL*EP**2.0*FMCEXP(-ALAM)*FMCDIF(HO,(X,2))-
    5.0E-1*AL*EP**2.0*FMCEXP(-ALAM-ANU)*FMCDIF(ALAM,(X,1))*FMCDIF(H1,(
    T,1))+AL*EP**2.0*FMCEXP(-ALAM-ANU)*FMCDIF(H1,(T,1),(X,1))-
    5.0E-1*AL*EP**2.0*FMCEXP(-ANU)*FMCDIF(AK,(T,2))-5.0E-1*AL*EP**2.0*
    FMCEXP(-ANU)*FMCDIF(H2,(T,2))-5.0E-1*EP**2.0*HO*X**(-2.0)*FMCSIN(Y
    )**(-1.0)*FMCCOS(Y)*FMCDIF(AL,(Y,1))+5.0E-1*EP**2.0*H2*X**(-2.0)*
    FMCSIN(Y)**(-1.0)*FMCCOS(Y)*FMCDIF(AL,(Y,1))+5.0E-1*X**(-1.0)*
    FMCEXP(-ALAM)*FMCDIF(ALAM,(X,1))-5.0E-1*X**(-1.0)*FMCEXP(-ALAM)*
    FMCDIF(ANU,(X,1))+2.5E-1*FMCEXP(-ALAM)*FMCDIF(ALAM,(X,1))*FMCDIF(
    ANU,(X,1))-2.5E-1*FMCEXP(-ALAM)*FMCDIF(ANU,(X,1))**2.0-5.0E-1*
    FMCEXP(-ALAM)*FMCDIF(ANU,(X,2))$
```

THE EINSTEIN TENSOR (continued)

```
 3U     3L
  5.0E-1*AL*EP**2.0*H2*X**(-1.0)*FMCEXP(-ALAM)*FMCDIF(ALAM,(X,1))-
  5.0E-1*AL*EP**2.0*H2*X**(-1.0)*FMCEXP(-ALAM)*FMCDIF(ANU,(X,1))+
  2.5E-1*AL*EP**2.0*H2*FMCEXP(-ALAM)*FMCDIF(ALAM,(X,1))*FMCDIF(ANU,(
  X,1))-2.5E-1*AL*EP**2.0*H2*FMCEXP(-ALAM)*FMCDIF(ANU,(X,1))**2.0-
  5.0E-1*AL*EP**2.0*H2*FMCEXP(-ALAM)*FMCDIF(ANU,(X,2))+AL*EP**2.0*X
  **(-1.0)*FMCEXP(-ALAM)*FMCDIF(AK,(X,1))-5.0E-1*AL*EP**2.0*X**(-1.0
  )*FMCEXP(-ALAM)*FMCDIF(HO,(X,1))-5.0E-1*AL*EP**2.0*X**(-1.0)*
  FMCEXP(-ALAM)*FMCDIF(H2,(X,1))+AL*EP**2.0*X**(-1.0)*FMCEXP(-ALAM-
  ANU)*FMCDIF(H1,(T,1))-2.5E-1*AL*EP**2.0*FMCEXP(-ALAM)*FMCDIF(AK,(X
  ,1))*FMCDIF(ALAM,(X,1))+2.5E-1*AL*EP**2.0*FMCEXP(-ALAM)*FMCDIF(AK,
  (X,1))*FMCDIF(ANU,(X,1))+5.0E-1*AL*EP**2.0*FMCEXP(-ALAM)*FMCDIF(AK
  ,(X,2))+2.5E-1*AL*EP**2.0*FMCEXP(-ALAM)*FMCDIF(ALAM,(X,1))*FMCDIF(
  HO,(X,1))-5.0E-1*AL*EP**2.0*FMCEXP(-ALAM)*FMCDIF(ANU,(X,1))*FMCDIF
  (HO,(X,1))-2.5E-1*AL*EP**2.0*FMCEXP(-ALAM)*FMCDIF(ANU,(X,1))*
  FMCDIF(H2,(X,1))-5.0E-1*AL*EP**2.0*FMCEXP(-ALAM)*FMCDIF(HO,(X,2))-
  5.0E-1*AL*EP**2.0*FMCEXP(-ALAM-ANU)*FMCDIF(ALAM,(X,1))*FMCDIF(H1,(
  T,1))+AL*EP**2.0*FMCEXP(-ALAM-ANU)*FMCDIF(H1,(T,1),(X,1))-
  5.0E-1*AL*EP**2.0*FMCEXP(-ANU)*FMCDIF(AK,(T,2))-5.0E-1*AL*EP**2.0*
  FMCEXP(-ANU)*FMCDIF(H2,(T,2))-5.0E-1*EP**2.0*HO*X**(-2.0)*FMCDIF(
  AL,(Y,2))+5.0E-1*EP**2.0*H2*X**(-2.0)*FMCDIF(AL,(Y,2))+5.0E-1*X**(
  -1.0)*FMCEXP(-ALAM)*FMCDIF(ALAM,(X,1))-5.0E-1*X**(-1.0)*FMCEXP(-
  ALAM)*FMCDIF(ANU,(X,1))+2.5E-1*FMCEXP(-ALAM)*FMCDIF(ALAM,(X,1))*
  FMCDIF(ANU,(X,1))-2.5E-1*FMCEXP(-ALAM)*FMCDIF(ANU,(X,1))**2.0-
  5.0E-1*FMCEXP(-ALAM)*FMCDIF(ANU,(X,2))$
 4U     4L
  AK*AL*EP**2.0*X**(-2.0)+5.0E-1*AK*EP**2.0*X**(-2.0)*FMCSIN(Y)**(
  -1.0)*FMCCOS(Y)*FMCDIF(AL,(Y,1))+5.0E-1*AK*EP**2.0*X**(-2.0)*
  FMCDIF(AL,(Y,2))-AL*EP**2.0*H2*X**(-2.0)*FMCEXP(-ALAM)+AL*EP**2.0*
  H2*X**(-1.0)*FMCEXP(-ALAM)*FMCDIF(ALAM,(X,1))+3.0*AL*EP**2.0*X**(
  -1.0)*FMCEXP(-ALAM)*FMCDIF(AK,(X,1))-AL*EP**2.0*X**(-1.0)*FMCEXP(-
  ALAM)*FMCDIF(H2,(X,1))-5.0E-1*AL*EP**2.0*FMCEXP(-ALAM)*FMCDIF(AK,(
  X,1))*FMCDIF(ALAM,(X,1))+AL*EP**2.0*FMCEXP(-ALAM)*FMCDIF(AK,(X,2))
  +5.0E-1*EP**2.0*H2*X**(-2.0)*FMCSIN(Y)**(-1.0)*FMCCOS(Y)*FMCDIF(AL
  ,(Y,1))+5.0E-1*EP**2.0*H2*X**(-2.0)*FMCDIF(AL,(Y,2))-X**(-2.0)*
  FMCEXP(-ALAM)+X**(-2.0)+X**(-1.0)*FMCEXP(-ALAM)*FMCDIF(ALAM,(X,1))
  $
```

C. RIEMANN, the Program for Calculating Riemann Tensors

1. Input for RIEMANN

RIEMANN is the program which calculates the covariant components, R_{abcd}, of the Riemann tensor. The input for RIEMANN is identical to that for EINSTEIN: The user must supply ATØMIC and DEPEND statements, an EPTERM declaration, and LET definitions of the metric components G1(I,J) and G2(I,J). (See § III.B.1.) A listing of RIEMANN showing where to insert the required cards follows.

2. Complete Listing of RIEMANN

```
$IBFMC TWO     NODECK
        SYMARG
C
C         RIEMANN, A PROGRAM TO CALCULATE THE RIEMANN TENSOR
C
      ATOMIC   X,Y,Z,T,EP,ANU,ALAM            } User:  Replace these cards
      DEPEND (ANU,ALAM/X)                     } with your own input.
C
C         ATOMIC AND DEPEND STATEMENTS MUST BE SUPPLIED BY USER
C         IMMEDIATELY PRECEDING THIS COMMENT.
C         THE ATOMIC STATEMENT MUST LIST ALL FUNCTIONS AND INDEPENDENT
C         VARIABLES USED IN A PARTICULAR PROBLEM.
C         THE DEPEND STATEMENT DEFINES ALL THE FUNCTIONAL DEPENDENCIES
C         FOR THE PROBLEM, I.E. WHICH FUNCTIONS DEPEND ON WHICH VARIABLES.
C
        REAL MCS(4,4,4)
        LOGICAL Q
        DIMENSION CARD(12),LINE(12),CCS(4,4,4),G1(4,4),G2(4,4)
C
        INTEGER EPTERM
C
C         IF EPTERM = 0 TERMS CONTAINING ALL POWERS OF EXPANSION
C                       PARAMETER WILL BE RETAINED.
C         IF EPTERM .NE. 0 TERMS CONTAINING THE EXPANSION PARAMETER
C                       TO A POWER GREATER THAN 2 WILL BE DISCARDED.
C         A CARD CONTAINING THE INTEGER VALUE OF EPTERM MUST FOLLOW THIS
C                       COMMENT.
C                                               } User:  Replace this card
      EPTERM = 0                                } with your own input.
C
        DO 1  I  = 1,4
        DO 1  J  = 1,4
        LET G1(I,J) = 0.
        LET G2(I,J) = 0.
    1 CONTINUE
C
C         ALL ELEMENTS OF THE COVARIANT METRIC TENSOR, G1(I,J), AND THE
C         CONTRAVARIANT METRIC TENSOR, G2(I,J), HAVE BEEN SET EQUAL TO
C         ZERO.
C         USER MUST SUPPLY ''LET'' STATEMENTS DEFINING ALL NON ZERO
C         ELEMENTS OF THESE TENSORS IMMEDIATELY FOLLOWING THIS COMMENT.
C
C
      LET G1(1,1) = -FMCEXP(ALAM)             )
      LET G1(2,2) = -X**2                     )
      LET G1(3,3) = -(X*FMCSIN(Y))**2         )
      LET G1(4,4) =  FMCEXP(ANU)              } User:  Replace these cards
      LET G2(1,1) = -FMCEXP(-ALAM)            } with your own input.
      LET G2(2,2) = -1/X**2                   )
      LET G2(3,3) = -1/(X*FMCSIN(Y))**2       )
      LET G2(4,4) =  FMCEXP(-ANU)             )
```

```
C
C
C     THE COVARIANT METRIC TENSOR -- OUTPUT
C
      WRITE(6,501)
  501 FORMAT( 51H1COVARIANT METRIC TENSOR NON ZERO ELEMENTS   (INPUT)   )
      DO 2 I = 1,4
      DO 2 J = I,4
      LET Q = MATCH ID, G1(I,J),0.
      IF(Q) GO TO 2
      WRITE(6,800)I,J
  800 FORMAT(1H I5,1HL I5,1HL )
      BEGIN = 0.
   46 LET BEGIN = BCDCON G1(I,J),LINE,12
      WRITE(6,200)(LINE(L),L=2,12)
      IF(BEGIN.NE.O.)GO TO 46
    2 CONTINUE
C
C     THE CONTRAVARIANT METRIC TENSOR  -- OUTPUT
C
      WRITE(6,700)
  700 FORMAT(54HOCONTRAVARIANT METRIC TENSOR NON ZERO ELEMENTS (INPUT))
      DO 47 I=1,4
      DO 47 J = I,4
      LET Q = MATCH ID, G2(I,J),0.
      IF(Q)GO TO 47
      WRITE(6,500) I,J
  500 FORMAT(1H I5,1HU I5,1HU )
      BEGIN = 0.
   48 LET BEGIN = BCDCON G2(I,J),LINE,12
      WRITE(6,200)(LINE(L),L=2,12)
      IF(BEGIN.NE.O.)GO TO 48
   47 CONTINUE
C
C     THE COVARIANT CHRISTOFFEL SYMBCLS
C
      WRITE(6,201)
  201 FORMAT(34H1THE COVARIANT CHRISTOFFEL SYMBOLS )
      DO 3 I = 1,4
      DO 3 J = I,4
      DO 3 K = 1,4
      LET T1 = 0.
      GO TO (31,32,70,71),I
   31 LET T1 = FMCDIF(G1(J,K),X,1)
      GO TO 33
   32 LET T1 = FMCDIF(G1(J,K),Y,1)
      GO TO 33
   70 LET T1 = FMCDIF(G1(J,K),Z,1)
      GO TO 33
   71 LET T1 = FMCDIF(G1(J,K),T,1)
   33 CONTINUE
```

```
      LET T2 = 0.
      GO TO(35,36,72,73),J
   35 LET T2 = FMCDIF(G1(I,K),X,1)
      GO TO 34
   36 LET T2 = FMCDIF(G1(I,K),Y,1)
       GO TO 34
   72 LET T2 = FMCDIF(G1(I,K),Z,1)
       GO TO 34
   73 LET T2 = FMCDIF(G1(I,K),T,1)
   34 CONTINUE
      LET T3 = 0.
      GO TO (38,39,74,75),K
   38 LET T3 = FMCDIF(G1(I,J),X,1)
      GO TO 37
   39 LET T3 = FMCDIF(G1(I,J),Y,1)
      GO TO 37
   74 LET T3 = FMCDIF(G1(I,J),Z,1)
      GO TO 37
   75 LET T3 = FMCDIF(G1(I,J),T,1)
   37 LET CCS(I,J,K) = 0.5*(T1+T2-T3)
      LET CCS(J,I,K) = CCS(I,J,K)
      LET Q = MATCH ID, CCS(I,J,K),0.
      IF(Q)GO TO 3
      WRITE(6,203)I,J,K
  203 FORMAT(1H I5, 2HLS I3,2HLS I4,1HL)
      LET ANS = EXPAND CCS(I,J,K)
      LET ANS = ORDER ANS,INC,FUL
      BEGIN = 0.
    5 LET BEGIN = BCDCON ANS,LINE,12
      WRITE(6,200)(LINE(L),L=2,12)
  200 FORMAT(1H  5X 12A6)
      IF(BEGIN.NE.0.)GO TO 5
      ERASE ANS
    3 CONTINUE
        ERASE T1,T2,T3
C
C       THE MIXED CHRISTOFFEL SYMBOLS
C
      WRITE(6,205)
  205 FORMAT( 30H1THE MIXED CHRISTOFFEL SYMBOLS )
      DO 6 I = 1,4
      DO 6 J = 1,4
      DO 6 K = 1,4
      LET MCS(I,J,K) = 0.
      DO 7 L = 1,4
      LET TT = G2(K,L)*CCS(I,J,L)
      LET TT = EXPAND TT
      IF(EPTERM)506,507,506
  507 LET SUM = TT
      ERASE TT
      GO TO 514
  506 CONTINUE
       LET S1 = COEFF TT,EP**0,R
       LET S2 = COEFF TT,EP**1,R
       LET S3 = COEFF TT,EP**2,R
       LET SUM = S1 + S2*EP + S3*EP*EP
       ERASE TT
      ERASE S1,S2,S3
  514 CONTINUE
       LET MCS(I,J,K) = MCS(I,J,K) + SUM
       ERASE SUM
    7 CONTINUE
       LET MCS(I,J,K) = EXPAND MCS(I,J,K)
       LET MCS(J,I,K) = MCS(I,J,K)
       LET Q = MATCH ID, MCS(I,J,K),0.
       IF(Q)GO TO 6
      WRITE(6,769) I,J,K
  769 FORMAT(1H I5,2HLS I3,2HLS I4,1HU )
       LET ANS = ORDER MCS(I,J,K),INC,FUL
       BEGIN = 0.
    8 LET BEGIN = BCDCON ANS,LINE,12
       WRITE(6,200)(LINE(L),L=2,12)
       IF(BEGIN.NE.0.)GO TO 8
       ERASE ANS
    6 CONTINUE
C
C
C        THE RIEMANN TENSOR
C
C
       WRITE(6,215)
  215 FORMAT(41H1THE RIEMANN TENSOR, COVARIANT COMPONENTS   )
       DO 15 I = 1,4
      LL = I+1
       iF(LL.GT.4)LL=4
       DO 15 J =LL,4
       DO 17 K = I,4
      DO 17 L = J,4
      LET SUM = 0.
       DO 18 MU = 1,4
       LET TT    =      (MCS(J,K,MU)*CCS(I,L,MU)-MCS(J,L,MU)*CCS(I,K,MU)
     1 )
       LET TT = EXPAND TT
       IF(EPTERM)520,521,520
  521 LET SUM = SUM + TT
       GO TO 18
  520  CCNTINUE
       LET S1 = COEFF TT,EP**0,R
       LET S2 = COEFF TT,EP**1,R
       LET S3 = COEFF TT,EP**2,R
       ERASE TT
       LET SUM = SUM + S1 + S2*EP + S3*EP*EP
      ERASE S1,S2,S3
 18  CONTINUE
      LET D1 = 0.
      GO TO (41,42,80,81),K
 41  LET D1 = FMCDIF(CCS(J,L,I),X,1)
      GO TO 40
 42  LET D1 = FMCDIF(CCS(J,L,I),Y,1)
      GO TO 40
 80  LET D1 = FMCDIF(CCS(J,L,I),Z,1)
      GO TO 40
 81  LET D1 = FMCDIF(CCS(J,L,I),T,1)
 40  LET D2 = 0.
      GO TO (44,45,82,83),L
 44  LET D2 = FMCDIF(CCS(J,K,I),X,1)
      GO TO 43
 45  LET D2 = FMCDIF(CCS(J,K,I),Y,1)
      GO TO 43
 82  LET D2 = FMCDIF(CCS(J,K,I),Z,1)
      GO TO 43
 83  LET D2 = FMCDIF(CCS(J,K,I),T,1)
 43  CONTINUE
      LET DIFF = D1-D2
      ERASE D1,D2
      LET RMT = DIFF + SUM
      ERASE SUM,DIFF
      LET ANS = EXPAND RMT
      ERASE RMT
      LET Q = MATCH ID, ANS,0.
      IF(Q)GO TO 17
      WRITE(6,206)I,J,K,L
206  FORMAT(1H 5X I5,1HL I5,1HL I5,1HL I5,1HL )
      LET ANS = ORDER ANS,INC,FUL
      BEGIN = 0.
 25  LET BEGIN = BCDCON ANS,LINE,12
      WRITE(6,200)(LINE(IL),IL=2,12)
      IF(BEGIN.NE.O.)GO TO 25
      ERASE ANS
 17  CONTINUE
 15  CONTINUE
      STOP
     END
```

3. Output for RIEMANN

The printed output for RIEMANN includes: the metric components, g_{ij} and g^{ij}; the Christoffel symbols (eqs. (2) and (3)); and the covariant components of the Riemann curvature tensor

$$R_{abcd} = \Gamma_{bc}{}^{i}\,\Gamma_{adi} - \Gamma_{bd}{}^{i}\,\Gamma_{aci} + \Gamma_{bda,c} - \Gamma_{bca,d}\ . \qquad (8)$$

The format is similar to that for the program EINSTEIN, and error messages are identical to those for EINSTEIN. The computer prints only the nonzero components of the Riemann tensor; and of those components which can be obtained from each other by symmetry operations, it prints only one. For example, R_{1234} is printed if it is nonzero; but R_{2134}, R_{1243}, R_{2143}, R_{3412}, R_{4312}, R_{3421}, and R_{4312} are never printed because

$$R_{1234} = - R_{2134} = - R_{1243} = + R_{2143} = + R_{3412} = - R_{4312} = - R_{3421} = + R_{4312}.$$

As an example of output from a computation by RIEMANN, we give below the results for the static, spherically-symmetric metric of equation (1). The input which generated these results was given in § III.B.1 and also in the listing of RIEMANN in § III.C.2.

METRIC TENSOR COMPONENTS AND CHRISTOFFEL SYMBOLS

— output identical to that from EINSTEIN as given on page 23.

```
IHE RIEMANN TENSOR, CCVARIANT CCMPONENTS
      1L     2L     1L     2L
  -5.0E-1*X*FMCCIF(ALAM,(X,1))$
      1L     3L     1L     3L
  -5.0E-1*X*FMCSIN(Y)**2.C*FMCCIF(ALAM,(X,1))$
      1L     4L     1L     4L
  2.5E-1*FMCEXP(ANL)*FMCCIF(ALAM,(X,1))*FMCDIF(ANU,(X,1))-2.5E-1*
  FMCEXP(ANL)*FMCDIF(ANL,(X,1))**2.0-5.0E-1*FMCEXP(ANU)*FMCCIF(ANL,(
  X,2))$
      2L     3L     2L     3L
  X**2.0*FMCSIN(Y)**2.0*FMCEXP(-ALAM)-X**2.0*FMCSIN(Y)**2.0$
      2L     4L     2L     4L
  -5.0E-1*X*FMCEXP(-ALAM+ANL)*FMCCIF(ANU,(X,1))$
      3L     4L     3L     4L
  -5.0E-1*X*FMCSIN(Y)**2.0*FMCEXP(-ALAM+ANU)*FMCCIF(ANL,(X,1))$
```

4. Sample Problems for RIEMANN

Input and output for one sample problem, the static spherically symmetric metric of equation (1), were given in the last two sections. As a second example consider the metric (7) for a star in even-parity, nonradial pulsation. The input to RIEMANN for this metric is identical to the input to EINSTEIN (see page 25). The output from RIEMANN is partially identical to that from EINSTEIN (pages 26 - 28 are identical); but instead of outputting Ricci and Einstein tensors after the Christoffel symbols, RIEMANN outputs Riemann tensor components, of which we present only five:

```
THE RIEMANN TENSOR, COVARIANT COMPONENTS
          1L      2L      1L      2L
        5.0E-1*AK*AL*EP**2.0*X*FMCDIF(ALAM,(X,1))-AL*EP**2.0*X*FMCDIF(AK,(
        X,1))+5.0E-1*AL*EP**2.0*X*FMCDIF(H2,(X,1))+2.5E-1*AL*EP**2.0*X**
        2.0*FMCDIF(AK,(X,1))*FMCDIF(ALAM,(X,1))-5.0E-1*AL*EP**2.0*X**2.0*
        FMCDIF(AK,(X,2))-5.0E-1*EP**2.0*H2*FMCEXP(ALAM)*FMCDIF(AL,(Y,2))-
        5.0E-1*X*FMCDIF(ALAM,(X,1))$
          1L      2L      1L      4L
        -5.0E-1*EP**2.0*H1*X**(-1.0)*FMCDIF(AL,(Y,1))-2.5E-1*EP**2.0*H1*
        FMCDIF(AL,(Y,1))*FMCDIF(ALAM,(X,1))+2.5E-1*EP**2.0*H1*FMCDIF(AL,(Y
        ,1))*FMCDIF(ANU,(X,1))-5.0E-1*EP**2.0*FMCEXP(ALAM)*FMCDIF(AL,(Y,1)
        )*FMCDIF(H2,(T,1))+5.0E-1*EP**2.0*FMCDIF(AL,(Y,1))*FMCDIF(H1,(X,1)
        )$
          1L      2L      2L      4L
        5.0E-1*AL*EP**2.0*H1*X*FMCEXP(-ALAM)*FMCDIF(ANU,(X,1))+5.0E-1*AL*
        EP**2.0*X*FMCDIF(AK,(T,1))-5.0E-1*AL*EP**2.0*X*FMCDIF(H2,(T,1))-
        2.5E-1*AL*EP**2.0*X**2.0*FMCDIF(AK,(T,1))*FMCDIF(ANU,(X,1))+
        5.0E-1*AL*EP**2.0*X**2.0*FMCDIF(AK,(T,1),(X,1))+5.0E-1*EP**2.0*H1*
        FMCDIF(AL,(Y,2))$
          1L      2L      4L      2L
        -5.0E-1*AL*EP**2.0*H1*X*FMCEXP(-ALAM)*FMCDIF(ANU,(X,1))-5.0E-1*AL*
        EP**2.0*X*FMCDIF(AK,(T,1))+5.0E-1*AL*EP**2.0*X*FMCDIF(H2,(T,1))+
        2.5E-1*AL*EP**2.0*X**2.0*FMCDIF(AK,(T,1))*FMCDIF(ANU,(X,1))-
        5.0E-1*AL*EP**2.0*X**2.0*FMCDIF(AK,(T,1),(X,1))-5.0E-1*EP**2.0*H1*
        FMCDIF(AL,(Y,2))$
          1L      3L      1L      3L
        5.0E-1*AK*AL*EP**2.0*X*FMCSIN(Y)**2.0*FMCDIF(ALAM,(X,1))-AL*EP**
        2.0*X*FMCSIN(Y)**2.0*FMCDIF(AK,(X,1))+5.0E-1*AL*EP**2.0*X*FMCSIN(Y
        )**2.0*FMCDIF(H2,(X,1))+2.5E-1*AL*EP**2.0*X**2.0*FMCSIN(Y)**2.0*
        FMCDIF(AK,(X,1))*FMCDIF(ALAM,(X,1))-5.0E-1*AL*EP**2.0*X**2.0*
        FMCSIN(Y)**2.0*FMCDIF(AK,(X,2))-5.0E-1*EP**2.0*H2*FMCSIN(Y)*FMCCOS
        (Y)*FMCEXP(ALAM)*FMCDIF(AL,(Y,1))-5.0E-1*X*FMCSIN(Y)**2.0*FMCDIF(
        ALAM,(X,1))$
```

D. MOTION, the Program for Calculating the Divergence of the Stress-Energy Tensor

1. Input for MOTION

MOTION is the program which calculates the divergence of the stress-energy tensor, $W_a{}^b{}_{;b}$. The input for MOTION is identical to that for EINSTEIN (page 9) except for this: One must also supply

6. "LET" statements defining all nonzero mixed components of the stress-energy tensor, $W_i{}^j$, in terms of atomic variables. $W_i{}^j$ is denoted in MOTION by W(I,J) – i.e., the first index is covariant; the second is contravariant.

A listing of MOTION showing where to insert the required cards follows:

2. Complete Listing of MOTION

```
$IBFMC EOM     NODECK
C
C          MOTION, A PROGRAM TO CALCULATE THE DIVERGENCE OF THE
C          STRESS-ENERGY TENSOR
C
C
         SYMARG
C
        ATOMIC X,Y,Z,T,EP,ANU,ALAM,P,RHO       } User:  Replace these cards
        DEPEND (ALAM,ANU,P,RHO/X)              } with your own input.
C
C
C          ATOMIC AND DEPEND STATEMENTS MUST BE SUPPLIED BY USER
C          IMMEDIATELY PRECEDING THIS COMMENT.
C          THE ATOMIC STATEMENT MUST LIST ALL FUNCTIONS AND INDEPENDENT
C          VARIABLES USED IN A PARTICULAR PROBLEM.
C          THE DEPEND STATEMENT DEFINES ALL THE FUNCTIONAL DEPENDENCIES
C          FOR THE PROBLEM, I.E. WHICH FUNCTIONS DEPEND ON WHICH VARIABLES.
C
         REAL MCS(4,4,4)
         LOGICAL Q
         DIMENSION CARD(12),LINE(12),CCS(4,4,4),G1(4,4),G2(4,4)
         DIMENSION W(4,4)
C
         INTEGER EPTERM
C
C          IF EPTERM = 0 TERMS CONTAINING ALL POWERS OF EXPANSION
C                        PARAMETER WILL BE RETAINED.
C          IF EPTERM .NE. 0 TERMS CONTAINING THE EXPANSION PARAMETER
C                        TO A POWER GREATER THAN 2 WILL BE DISCARDED.
C          A CARD CONTAINING THE INTEGER VALUE OF EPTERM MUST FOLLOW
C                        THIS COMMENT.
C
                                               } User:  Replace this card
        EPTERM = 0                             } with your own input.
C
         DO 1 I = 1,4
         DO 1 J = 1,4
         LET G1(I,J) = 0.
        LET G2(I,J) = 0.
     1  CONTINUE
C
C          ALL ELEMENTS OF THE COVARIANT METRIC TENSOR, G1(I,J), AND THE
C          CONTRAVARIANT METRIC TENSOR, G2(I,J), HAVE BEEN SET EQUAL
C          TO ZERO.
C          USER MUST SUPPLY ''LET'' STATEMENTS DEFINING ALL NON ZERO
C          ELEMENTS OF THESE TENSORS IMMEDIATELY FOLLOWING THIS COMMENT.
C
C
        LET G1(1,1) = -FMCEXP(ALAM)            }
        LET G1(2,2) = -X**2                    }
        LET G1(3,3) = -(X*FMCSIN(Y))**2        }
        LET G1(4,4) =  FMCEXP(ANU)             } User:  Replace these cards
        LET G2(1,1) = -FMCEXP(-ALAM)           } with your own input.
        LET G2(2,2) = -1/X**2                  }
        LET G2(3,3) = -1/(X*FMCSIN(Y))**2      }
        LET G2(4,4) =  FMCEXP(-ANU)            }
```

```
C
       WRITE(6,501)
  501 FORMAT( 51H1COVARIANT METRIC TENSOR NON ZERU ELEMENTS  (INPUT)   )
       DO 2 I = 1,4
       DO 2 J = 1,4
       LET Q = MATCH ID, G1(I,J),0.
       IF(Q) GO TO 2
       WRITE(6,800)I,J
  800 FORMAT(1H I5,1HL I5,1HL )
       BEGIN = 0.
   40 LET BEGIN = BCDCON G1(I,J),LINE,12
       WRITE(6,200)(LINE(L),L=2,12)
       IF(BEGIN.NE.0.)GO TU 40
    2 CONTINUE
      WRITE(6,700)
  700 FORMAT(54H0CONTRAVARIANT METRIC TENSUR NON ZERO ELEMENTS (INPUT))
       DO 42 I = 1,4
       DO 42 J = 1,4
       LET Q = MATCH ID, G2(I,J),0.
       IF(Q)GO TO 42
      WRITE(6,500) I,J
  500 FORMAT(1H I5,1HU I5,1HU )
      BEGIN = 0.
   41 LET BEGIN = BCDCON G2(I,J),LINE,12
       WRITE(6,200)(LINE(L),L=2,12)
       IF(BEGIN.NE.0.)GO TO 41
   42 CONTINUE
       DO 20 I = 1,4
       DO 20 J = 1,4
       LET W(I,J) = 0.
   20 CONTINUE
C
C        ALL ELEMENTS OF THE STRESS-ENERGY TENSOR, MIXED COMPCNENTS
C        W(I,J), ARE SET EQUAL TO ZERO.   NOTE---THE FIRST INDEX OF W
C        IS DOWN (COVARIANT) AND THE SECONC INDEX IS UP (CONTRAVARIANT).
C        USER MUST SUPPLY ''LET'' STATEMENTS DEFINING ALL NON ZERO
C        ELEMENTS IMMEDIATELY FOLLOWING THIS COMMENT
C
      LET W(1,1) = -P                  }
      LET W(2,2) = -P                  }  User:  Replace these cards
      LET W(3,3) = -P                  }  with your own input.
      LET W(4,4) =  RHO                }
C
      WRITE(6,208)
  208 FORMAT(65H1STRESS-ENERGY TENSOR NON ZERO ELEMENTS, MIXEC CCMPONENT
     1S (INPUT)    )
```

```
      DO 16 I=1,4
      DO 16 J = 1,4
      LET Q = MATCH ID,W(I,J),0.
      IF(Q)GO TO 16
     WRITE(6,505) I,J
505  FORMAT(1H I5, 1HL I4, 1HU )
       BEGIN = 0.
      LET ANS=W(I,J)
 96    LET BEGIN = BCDCON ANS    ,LINE,12
      WRITE(6,200)(LINE(L),L=2,12)
      IF(BEGIN.NE.0.) GO TO 96
     ERASE ANS
 16 CONTINUE
     WRITE(6,878)
878 FORMAT( 35H1THE COVARIANT CHRISTOFFEL SYMBOLS  )
      DO 3 I = 1,4
      DO 3 J = 1,4
      DO 3 K = 1,4
      LET T1 = 0.
      GO TO (31,32,70,71),I
 31  LET T1 = FMCDIF(G1(J,K),X,1)
      GO TO 33
 32  LET T1 = FMCDIF(G1(J,K),Y,1)
      GO TO 33
 70  LET T1 = FMCDIF(G1(J,K),Z,1)
      GO TO 33
 71  LET T1 = FMCDIF(G1(J,K),T,1)
 33  CONTINUE
      LET T2 = 0.
      GO TO(35,36,72,73),J
 35  LET T2 = FMCDIF(G1(I,K),X,1)
      GO TO 34
 36  LET T2 = FMCDIF(G1(I,K),Y,1)
       GO TO 34
 72  LET T2 = FMCDIF(G1(I,K),Z,1)
       GO TO 34
 73  LET T2 = FMCDIF(G1(I,K),T,1)
 34  CONTINUE
      LET T3 = 0.
      GO TO (38,39,74,75),K
 38  LET T3 = FMCDIF(G1(I,J),X,1)
      GO TO 37
 39  LET T3 = FMCDIF(G1(I,J),Y,1)
      GO TO 37
 74  LET T3 = FMCDIF(G1(I,J),Z,1)
      GO TO 37
 75  LET T3 = FMCDIF(G1(I,J),T,1)
 37  LET CCS(I,J,K) = 0.5*(T1+T2-T3)
      LET CCS(J,I,K) = CCS(I,J,K)
      LET Q = MATCH ID, CCS(I,J,K),0.
      IF(Q)GO TO 3
      WRITE(6,203)I,J,K
  203 FORMAT(1H I5,2HLS I3,2HLS I4,1HL)
      LET ANS = EXPAND CCS(I,J,K)
      LET ANS = ORDER ANS,INC,FUL
      BEGIN = 0.
    5 LET BEGIN = BCDCON ANS,LINE,12
      WRITE(6,200)(LINE(L),L=2,12)
  200 FORMAT(1H  5X 12A6)
      IF(BEGIN.NE.0.)GO TO 5
      ERASE ANS
    3 CONTINUE
      ERASE T1,T2,T3
      WRITE(6,209)
  209 FORMAT(1H0)
      WRITE(6,210)
  210 FORMAT(28H0ALL OTHER COMPONENTS VANISH   )
      WRITE(6,205)
  205 FORMAT( 30H1THE MIXED CHRISTOFFEL SYMBOLS )
      DO 6 I = 1,4
      DO 6 J = I,4
      DO 6 K = 1,4
      LET MCS(I,J,K) = 0.
      DO 7 L = 1,4
      LET TT = G2(K,L)*CCS(I,J,L)
      LET TT = EXPAND TT
      LET SUM = TT
      IF(EPTERM.EQ.0)GO TO 502
      LET S1 = COEFF TT,EP**0,R
      LET S2 = COEFF TT,EP**1,R
      LET S3 = COEFF TT,EP**2,R
      ERASE TT
      LET SUM = S1 + S2*EP + S3*EP*EP
      ERASE S1,S2,S3
  502 LET MCS(I,J,K) = MCS(I,J,K) + SUM
      ERASE SUM
    7 CONTINUE
      LET MCS(I,J,K) = EXPAND MCS(I,J,K)
      LET MCS(J,I,K) = MCS(I,J,K)
      LET Q = MATCH ID, MCS(I,J,K),0.
      IF(Q)GO TO 6
      WRITE(6,204)I,J,K
  204 FORMAT(1H I5,2HLS I3,2HLS I4,1HU )
      LET ANS = ORDER MCS(I,J,K),INC,FUL
      BEGIN = 0.
    8 LET BEGIN = BCDCON ANS,LINE,12
      WRITE(6,200)(LINE(L),L=2,12)
      IF(BEGIN.NE.0.)GO TO 8
      ERASE ANS
    6 CONTINUE
      DO 350 I = 1,4
      DO 350 J = 1,4
      DO 350 K = 1,4
      ERASE CCS(I,J,K)
  350 CONTINUE
      WRITE(6,209)
      WRITE(6,210)
C
C        THE EQUATIONS OF MOTION EM(MU)
C
      WRITE(6,207)
  207 FORMAT(47H1THE EQUATIONS OF MOTION, COVARIANT COMPCNENTS  )
      DO 21 MU = 1,4
      LET S1 = 0.
      DO 22 NU = 1,4
      LET T1 = 0.
      GO TO (23,24,25,26),NU
   23 LET T1 = FMCDIF(W(MU,NU),X,1)
      GO TO 27
   24 LET T1 = FMCDIF(W(MU,NU),Y,1)
      GO TO 27
   25 LET T1 = FMCDIF(W(MU,NU),Z,1)
      GO TO 27
   26 LET T1 = FMCDIF(W(MU,NU),T,1)
   27 CONTINUE
      LET S1 = S1 + T1
      LET S1 = EXPAND S1
   22 CONTINUE
      LET S2 = 0.
      LET S3 = 0.
      DO 28 I = 1,4
      DO 28 NU = 1,4
      LET T1 = MCS(NU,I,NU)*W(MU,I)
      LET T1 = EXPAND T1
      IF(EPTERM)508,509,508
  509 LET S2 = S2 + T1
       ERASE T1
      GO TO 512
  508 CONTINUE
      LET S0 = COEFF T1,EP**0,R
      LET S4 = COEFF T1,EP**1,R
      LET S5 = COEFF T1,EP**2,R
      ERASE T1
      LET S2 = S2 + (S0 + S4*EP + S5*EP*EP)
      ERASE S0,S4,S5
  512 CONTINUE
      LET S2 = EXPAND S2
      LET T2 = MCS(NU,MU,I)*W(I,NU)
      LET T2 = EXPAND T2
      IF(EPTERM)511,510,511
  510 LET S3 = S3 + T2
      ERASE T2
      LET S3 = EXPAND S3
      GO TO 28
  511 CONTINUE
```

```
      LET S0 = COEFF T2,EP**0,R
      LET S4 = COEFF T2,EP**1,R
      LET S5 = COEFF T2,EP**2,R
      ERASE T2
      LET S3 = S3 + (S0 + S4*EP + S5*EP*EP)
      ERASE S0,S4,S5
      LET S3 = EXPAND S3
  28 CONTINUE
      LET EM      = S1+S2-S3
      ERASE S1,S2,S3
      WRITE(6,206)MU
 206 FORMAT(1H0 I5, 1HL )
 563 FORMAT(I5)
      LET ANS = EXPAND EM
      ERASE EM
      LET ANS = ORDER ANS,INC,FUL
       BEGIN = 0.
  29 LET BEGIN = BCDCON ANS,LINE,12
      WRITE(6,200)(LINE(IL),IL=2,12)
 562 FORMAT(12A6)
      IF(BEGIN.NE.0.)GO TO 29
      ERASE ANS
  21 CONTINUE
2500 CONTINUE
      STOP
      END
```

3. Output for MOTION

The printed output for MOTION includes: the metric components, g_{ij} and g^{ij}, and the stress-energy tensor, $W_i^{\ j}$ – all of which are input by the user –; the Christoffel symbols (eqs. (2) and (3)); and the divergence of the stress-energy tensor ("equations of motion")

$$(EM)_a = W_a^{\ i}{}_{;i} = W_a^{\ i}{}_{,i} + \Gamma_{ji}^{\ \ i} W_a^{\ j} - \Gamma_{ai}^{\ \ j} W_j^{\ i} . \qquad (9)$$

The format is similar to that of EINSTEIN; and error messages are identical to those for EINSTEIN.

As an example of output from a computation by MOTION, we give below the results for the interior of a static, relativistic stellar model. The metric is the static, spherically symmetric metric of equation (1), and the stress-energy tensor is

$$T_1^{\ 1} = T_2^{\ 2} = T_3^{\ 3} = -p, \quad T_4^{\ 4} = \rho .$$

Here p (pressure) and ρ (density of mass-energy) are functions of $r = X$ only. This metric and stress-energy tensor are precisely the ones included in the listing of MOTION in the last section. The output follows:

METRIC TENSOR COMPONENTS (INPUT)

– identical to those given for this metric on page 23.

```
STRESS-ENERGY TENSOR NON ZERO ELEMENTS, MIXED COMPONENTS (INPUT)
    1L     1U
     -P$
    2L     2U
     -P$
    3L     3U
     -P$
    4L     4U
     RHO$
```

CHRISTOFFEL SYMBOLS (OUTPUT)

– identical to those given for this metric on page 22.

```
THE EQUATIONS OF MOTION, COVARIANT COMPONENTS

    1L
     -5.0E-1*P*FMCDIF(ANU,(X,1))-5.0E-1*RHO*FMCDIF(ANU,(X,1))-FMCDIF(P,
     (X,1))$

    2L
     0.0$

    3L
     0.0$

    4L
     0.0$
```

4. Sample Problems for MOTION

Input and output for one sample problem, the interior of a static relativistic stellar model, were given in the last two sections. As a second example, consider a star in even-parity, nonradial pulsation. The metric for such a star was given in equation (7), and the stress-energy tensor is as follows (cf. Thorne and Campolattaro 1967):

$$\begin{aligned}
W_1{}^1 &= W_2{}^2 = W_3{}^3 = -\,p + r^{-2}\,e^{-\lambda/2}\,W\,P_\ell(\partial p/\partial r) - \gamma p(\Delta n/n),\\
W_4{}^4 &= \rho - r^{-2}\,e^{-\lambda/2}\,W\,P_\ell(\partial\rho/\partial r) + (\rho+p)(\Delta n/n)\\
W_1{}^4 &= (\rho+p)\,e^{-\nu}\,P_\ell\left[H_1 - r^{-2}\,e^{\lambda/2}\,(\partial W/\partial t)\right],\\
W_4{}^1 &= (\rho+p)\,r^{-2}\,e^{-\lambda/2}\,P_\ell(\partial W/\partial t),\\
W_2{}^4 &= (\rho+p)\,e^{-\nu}\,(\partial P_\ell/\partial\theta)(\partial V/\partial t),\\
W_4{}^2 &= -\,(\rho+p)\,r^{-2}\,(\partial P_\ell/\partial\theta)(\partial V/\partial t).
\end{aligned}\tag{10}$$

Here ρ (density), p (pressure), and γ (adiabatic index) are functions of r only; $\Delta n/n$ is defined by

$$(\Delta n/n) = \left[-\,r^{-2}\,e^{-\lambda/2}(\partial W/\partial r) - \ell(\ell+1)\,r^{-2}\,V + K + H_2/2\right]P_\ell; \tag{11}$$

ℓ is the integer order of $P_\ell(\cos\theta)$; W and V are functions of r and t associated with the displacement of the fluid from equilibrium; and the remaining quantities were discussed on pages 25 and 26.

To prepare input for computing the divergence of the stress-energy tensor, we make the following changes to FØRMAC notation in addition to those discussed on page 25 :

$\ell \rightarrow$ AJ,	$p \rightarrow$ P
$V \rightarrow$ V,	$\rho \rightarrow$ RHØ
$W \rightarrow$ WF,	$\gamma \rightarrow$ GAM

Also, because we need answers only to first order in the perturbation from equilibrium, we attach the expansion parameter EP**2 to the perturbation quantities AK, H0, H1, H2, V, and WF, wherever they appear. The resultant input and output are as follows:

INPUT:

```
ATOMIC X,Y,Z,T,EP,AL,AJ,ANU,ALAM,AK,HO,H1,H2,V,WF,GAM,P,RHO
DEPEND (ANU,ALAM,GAM,P,RHO/X),(AL/Y)
DEPEND (HO,H1,H2,V,WF,AK/X,T)



 EPTERM = 3


LET G1(1,1) = -(-AL*EP**2*H2+1.)*FMCEXP(ALAM)
LET G1(1,4) =  AL*EP**2*H1
LET G1(4,1) =  G1(1,4)
LET G1(2,2) = -(1.-AK*AL*EP**2)*X**2
LET G1(3,3) = -(1.-AK*AL*EP**2)*X**2*FMCSIN(Y)**2
LET G1(4,4) =  (AL*EP**2*HO+1.)*FMCEXP(ANU)
LET G2(1,1) = -(AL*EP**2*H2+1.)*FMCEXP(-ALAM)
LET G2(1,4) =  AL*EP**2*H1*FMCEXP(-ALAM-ANU)
LET G2(4,1) =  G2(1,4)
LET G2(2,2) = -(AK*AL*EP**2+1.)*X**(-2)
LET G2(3,3) = -(AK*AL*EP**2+1.)*(X*FMCSIN(Y))**(-2)
LET G2(4,4) =  (-AL*EP**2*HO+1.)*FMCEXP(-ANU)




  LET W(1,1) = -(-AJ*(AJ+1.)*V*X**(-2)+AK+H2/2.-X**(-2)*
1                FMCEXP(-ALAM/2.)*FMCDIF(WF,(X,1)))*AL*EP**2*GAM*
2                P+AL*EP**2 *WF*X**(-2)*FMCEXP(-ALAM/2.)*
3                FMCDIF(P,(X,1))-P
  LET W(1,4) =   AL*EP**2*H1*(P+RHO)*FMCEXP(-ANU)-AL*EP**2*
1                (P+RHO)*X**(-2)*FMCEXP(ALAM/2.-ANU)*
2                FMCDIF(WF,(T,1))
  LET W(2,2) =   W(1,1)
  LET W(2,4) =   EP**2*(P+RHO)*FMCEXP(-ANU)*FMCDIF(AL,(Y,1))*
1                FMCDIF(V,(T,1))
  LET W(3,3) =   W(1,1)
  LET W(4,1) =   AL*EP**2*(P+RHO)*X**(-2)*FMCEXP(-ALAM/2.)*
1                FMCDIF(WF,(T,1))
  LET W(4,2) = -EP**2*(P+RHO)*X**(-2)*FMCDIF(AL,(Y,1))*
1                FMCDIF(V,(T,1))
  LET W(4,4) =   (-AJ*(AJ+1.)*V*X**(-2)+AK+H2/2.-X**(-2)*
1                FMCEXP(-ALAM/2.)*FMCDIF(WF,(X,1)))*AL*EP**2*
2                (P+RHO)-AL*EP**2*WF*X**(-2)*FMCEXP(-ALAM/2.)*
3                FMCDIF(RHO,(X,1))+RHO
```

OUTPUT:

```
   COVARIANT METRIC TENSOR NON ZERO ELEMENTS  (INPUT)
        1L      1L
         -(-AL*EP**2.0*H2+1.C)*FMCEXP(ALAM)$
        1L      4L
         AL*EP**2.0*H1$
        2L      2L
         -(-AK*AL*EP**2.0+1.C)*X**2.0$
        3L      3L
         -(-AK*AL*EP**2.0+1.C)*X**2.0*FMCSIN(Y)**2.0$
        4L      4L
         (AL*EP**2.0*H0+1.0)*FMCEXP(ANU)$

   CONTRAVARIANT METRIC TENSOR NON ZERO ELEMENTS (INPUT)
        1U      1U
         -(AL*EP**2.0*H2+1.0)*FMCEXP(-ALAM)$
        1U      4U
         AL*EP**2.0*H1*FMCEXP(-ALAM-ANU)$
        2U      2U
         -(AK*AL*EP**2.0+1.0)*X**(-2.0)$
        3U      3U
         -(AK*AL*EP**2.0+1.0)*X**(-2.0)*FMCSIN(Y)**(-2.0)$
        4U      4U
         (-AL*EP**2.0*H0+1.0)*FMCEXP(-ANU)$




   STRESS-ENERGY TENSOR NON ZERO ELEMENTS, MIXED COMPONENTS (INPUT)
        1L     1U
         -(-AJ*(AJ+1.0)*V*X**(-2.0)+AK+H2*5.0E-1-X**(-2.0)*FMCEXP(ALAM*(
         -5.0E-1))*FMCDIF(WF,(X,1)))*AL*EP**2.0*GAM*P+AL*EP**2.0*WF*X**(
         -2.0)*FMCEXP(ALAM*(-5.0E-1))*FMCDIF(P,(X,1))-P$
        1L     4U
         AL*EP**2.0*H1*(P+RHC)*FMCEXP(-ANU)-AL*EP**2.0*(P+RHO)*X**(-2.0)*
         FMCEXP(ALAM*5.0E-1-ANU)*FMCDIF(WF,(T,1))$
        2L     2U
         -(-AJ*(AJ+1.0)*V*X**(-2.0)+AK+H2*5.0E-1-X**(-2.0)*FMCEXP(ALAM*(
         -5.0E-1))*FMCDIF(WF,(X,1)))*AL*EP**2.0*GAM*F+AL*EP**2.0*WF*X**(
         -2.0)*FMCEXP(ALAM*(-5.0E-1))*FMCDIF(P,(X,1))-P$
        2L     4U
         EP**2.0*(P+RHO)*FMCEXP(-ANU)*FMCDIF(AL,(Y,1))*FMCCIF(V,(T,1))$
        3L     3U
         -(-AJ*(AJ+1.0)*V*X**(-2.0)+AK+H2*5.0E-1-X**(-2.0)*FMCEXP(ALAM*(
         -5.0E-1))*FMCDIF(WF,(X,1)))*AL*EP**2.0*GAM*P+AL*EP**2.0*WF*X**(
         -2.0)*FMCEXP(ALAM*(-5.0E-1))*FMCDIF(P,(X,1))-P$
        4L     1U
         AL*EP**2.0*(P+RHO)*X**(-2.0)*FMCEXP(ALAM*(-5.0E-1))*FMCCIF(WF,(T,1
         ))$
        4L     2U
         -EP**2.0*(P+RHO)*X**(-2.0)*FMCDIF(AL,(Y,1))*FMCDIF(V,(T,1))$
        4L     4U
         (-AJ*(AJ+1.0)*V*X**(-2.0)+AK+H2*5.0E-1-X**(-2.0)*FMCEXP(ALAM*(
         -5.0E-1))*FMCDIF(WF,(X,1)))*AL*EP**2.0*(P+RHO)-AL*EP**2.0*WF*X**(
         -2.0)*FMCEXP(ALAM*(-5.0E-1))*FMCDIF(RHO,(X,1))+RHC$
```

CHRISTOFFEL SYMBOLS

- identical to those on pages 27-29.

THE EQUATIONS OF MOTION, COVARIANT COMPONENTS

```
1L
 -2.0*AJ*AL*EP**2.0*GAM*P*V*X**(-3.0)+5.0E-1*AJ*AL*EP**2.0*GAM*P*V*
 X**(-2.0)*FMCDIF(ANU,(X,1))+AJ*AL*EP**2.0*GAM*P*X**(-2.0)*FMCDIF(V
 ,(X,1))+AJ*AL*EP**2.0*GAM*V*X**(-2.0)*FMCDIF(P,(X,1))+5.0E-1*AJ*AL
 *EP**2.0*P*V*X**(-2.0)*FMCDIF(ANU,(X,1))+AJ*AL*EP**2.0*P*V*X**(
 -2.0)*FMCDIF(GAM,(X,1))+5.0E-1*AJ*AL*EP**2.0*RHO*V*X**(-2.0)*
 FMCDIF(ANU,(X,1))-2.0*AJ**2.0*AL*EP**2.0*GAM*P*V*X**(-3.0)+
 5.0E-1*AJ**2.0*AL*EP**2.0*GAM*P*V*X**(-2.0)*FMCDIF(ANU,(X,1))+AJ**
 2.0*AL*EP**2.0*GAM*P*X**(-2.0)*FMCDIF(V,(X,1))+AJ**2.0*AL*EP**2.0*
 GAM*V*X**(-2.0)*FMCDIF(P,(X,1))+5.0E-1*AJ**2.0*AL*EP**2.0*P*V*X**(
 -2.0)*FMCDIF(ANU,(X,1))+AJ**2.0*AL*EP**2.0*P*V*X**(-2.0)*FMCDIF(
 GAM,(X,1))+5.0E-1*AJ**2.0*AL*EP**2.0*RHO*V*X**(-2.0)*FMCCIF(ANU,(X
 ,1))-5.0E-1*AK*AL*EP**2.0*GAM*P*FMCDIF(ANU,(X,1))-AK*AL*EP**2.0*
 GAM*FMCDIF(P,(X,1))-5.0E-1*AK*AL*EP**2.0*P*FMCDIF(ANU,(X,1))-AK*AL
 *EP**2.0*P*FMCDIF(GAM,(X,1))-5.0E-1*AK*AL*EP**2.0*RHO*FMCDIF(ANU,(
 X,1))-2.5E-1*AL*EP**2.0*GAM*H2*P*FMCDIF(ANU,(X,1))-5.0E-1*AL*EP**
 2.0*GAM*H2*FMCDIF(P,(X,1))-2.0*AL*EP**2.0*GAM*P*X**(-3.0)*FMCEXP(-
 5.0E-1*ALAM)*FMCDIF(WF,(X,1))-5.0E-1*AL*EP**2.0*GAM*P*X**(-2.0)*
 FMCEXP(-5.0E-1*ALAM)*FMCDIF(ALAM,(X,1))*FMCDIF(WF,(X,1))+
 5.0E-1*AL*EP**2.0*GAM*P*X**(-2.0)*FMCEXP(-5.0E-1*ALAM)*FMCDIF(ANU,
 (X,1))*FMCDIF(WF,(X,1))+AL*EP**2.0*GAM*P*X**(-2.0)*FMCEXP(-
 5.0E-1*ALAM)*FMCDIF(WF,(X,2))-AL*EP**2.0*GAM*P*FMCCIF(AK,(X,1))-
 5.0E-1*AL*EP**2.0*GAM*P*FMCDIF(H2,(X,1))+AL*EP**2.0*GAM*X**(-2.0)*
 FMCEXP(-5.0E-1*ALAM)*FMCDIF(P,(X,1))*FMCDIF(WF,(X,1))-2.5E-1*AL*EP
 **2.0*H2*P*FMCDIF(ANU,(X,1))-5.0E-1*AL*EP**2.0*H2*P*FMCCIF(GAM,(X,
 1))-2.5E-1*AL*EP**2.0*H2*RHO*FMCDIF(ANU,(X,1))+5.0E-1*AL*EP**2.0*P
 *X**(-2.0)*FMCEXP(-5.0E-1*ALAM)*FMCDIF(ANU,(X,1))*FMCDIF(WF,(X,1))
 +AL*EP**2.0*P*X**(-2.0)*FMCEXP(-5.0E-1*ALAM)*FMCCIF(GAM,(X,1))*
 FMCDIF(WF,(X,1))-AL*EP**2.0*P*X**(-2.0)*FMCEXP(5.0E-1*ALAM-ANU)*
 FMCDIF(WF,(T,2))+AL*EP**2.0*P*FMCEXP(-ANU)*FMCDIF(H1,(T,1))-
 5.0E-1*AL*EP**2.0*P*FMCDIF(H0,(X,1))+5.0E-1*AL*EP**2.0*RHO*X**(
 -2.0)*FMCEXP(-5.0E-1*ALAM)*FMCDIF(ANU,(X,1))*FMCDIF(WF,(X,1))-AL*
 EP**2.0*RHO*X**(-2.0)*FMCEXP(5.0E-1*ALAM-ANU)*FMCDIF(WF,(T,2))+AL*
 EP**2.0*RHO*FMCEXP(-ANU)*FMCDIF(H1,(T,1))-5.0E-1*AL*EP**2.0*RHO*
 FMCDIF(H0,(X,1))-2.0*AL*EP**2.0*WF*X**(-3.0)*FMCEXP(-5.0E-1*ALAM)*
 FMCDIF(P,(X,1))-5.0E-1*AL*EP**2.0*WF*X**(-2.0)*FMCEXP(-5.0E-1*ALAM
 )*FMCDIF(ALAM,(X,1))*FMCDIF(P,(X,1))+5.0E-1*AL*EP**2.0*WF*X**(-2.0
 )*FMCEXP(-5.0E-1*ALAM)*FMCDIF(ANU,(X,1))*FMCDIF(P,(X,1))+
 5.0E-1*AL*EP**2.0*WF*X**(-2.0)*FMCEXP(-5.0E-1*ALAM)*FMCCIF(ANU,(X,
 1))*FMCDIF(RHO,(X,1))+AL*EP**2.0*WF*X**(-2.0)*FMCEXP(-5.0E-1*ALAM)
 *FMCDIF(P,(X,2))+AL*EP**2.0*X**(-2.0)*FMCEXP(-5.0E-1*ALAM)*FMCDIF(
 P,(X,1))*FMCDIF(WF,(X,1))-5.0E-1*P*FMCDIF(ANU,(X,1))-5.0E-1*RHO*
 FMCDIF(ANU,(X,1))-FMCDIF(P,(X,1))$
```

THE EQUATIONS OF MOTION (continued)

```
2L
 AJ*EP**2.0*GAM*P*V*X**(-2.0)*FMCDIF(AL,(Y,1))+AJ**2.0*EP**2.0*GAM*
 P*V*X**(-2.0)*FMCDIF(AL,(Y,1))-AK*EP**2.0*GAM*P*FMCDIF(AL,(Y,1))-
 5.0E-1*EP**2.0*GAM*H2*P*FMCDIF(AL,(Y,1))+EP**2.0*GAM*P*X**(-2.0)*
 FMCEXP(-5.0E-1*ALAM)*FMCDIF(AL,(Y,1))*FMCDIF(WF,(X,1))-5.0E-1*EP**
 2.0*HO*P*FMCDIF(AL,(Y,1))-5.0E-1*EP**2.0*HO*RHO*FMCDIF(AL,(Y,1))+
 EP**2.0*P*FMCEXP(-ANU)*FMCDIF(AL,(Y,1))*FMCDIF(V,(T,2))+EP**2.0*
 RHO*FMCEXP(-ANU)*FMCDIF(AL,(Y,1))*FMCDIF(V,(T,2))+EP**2.0*WF*X**(
 -2.0)*FMCEXP(-5.0E-1*ALAM)*FMCDIF(AL,(Y,1))*FMCDIF(P,(X,1))$

3L
 0.0$

4L
 -AJ*AL*EP**2.0*P*X**(-2.0)*FMCDIF(V,(T,1))-AJ*AL*EP**2.0*RHO*X**(
 -2.0)*FMCDIF(V,(T,1))-AJ**2.0*AL*EP**2.0*P*X**(-2.0)*FMCDIF(V,(T,1
 ))-AJ**2.0*AL*EP**2.0*RHO*X**(-2.0)*FMCDIF(V,(T,1))+5.0E-1*AL*EP**
 2.0*P*X**(-2.0)*FMCEXP(-5.0E-1*ALAM)*FMCDIF(ANU,(X,1))*FMCDIF(WF,(
 T,1))+5.0E-1*AL*EP**2.0*RHO*X**(-2.0)*FMCEXP(-5.0E-1*ALAM)*FMCDIF(
 ANU,(X,1))*FMCDIF(WF,(T,1))+AL*EP**2.0*X**(-2.0)*FMCEXP(-
 5.0E-1*ALAM)*FMCDIF(P,(X,1))*FMCDIF(WF,(T,1))-EP**2.0*P*X**(-2.0)*
 FMCSIN(Y)**(-1.0)*FMCCOS(Y)*FMCDIF(AL,(Y,1))*FMCDIF(V,(T,1))-EP**
 2.0*P*X**(-2.0)*FMCDIF(AL,(Y,2))*FMCDIF(V,(T,1))-EP**2.0*RHO*X**(
 -2.0)*FMCSIN(Y)**(-1.0)*FMCCOS(Y)*FMCDIF(AL,(Y,1))*FMCDIF(V,(T,1))
 -EP**2.0*RHO*X**(-2.0)*FMCDIF(AL,(Y,2))*FMCDIF(V,(T,1))$
```

BLANK PAGE

E. SUBSTITUTE, the Program for Making Changes of Variables

1. Input for SUBSTITUTE

SUBSTITUTE is the program which takes long, complicated expressions and makes changes of variables in them. The input for SUBSTITUTE is handled in a slightly different manner than that for EINSTEIN, RIEMANN, and MOTION: Four statements are supplied by the user in the body of the program, and the remaining input information is supplied on "data cards" at the end of the program.

The statements supplied in the body of the program are these:

1. ATØMIC and DEPEND statements. In these statements must be listed <u>all</u> atomic variables for the original, unsubstituted expressions <u>and</u> for the substitutions which are to be made. For example, if $\lambda(X)$ in the expression

$$G_4{}^4 \equiv e^{-\lambda}\left(-\frac{1}{X^2}+\frac{\partial\lambda/\partial X}{X}\right)+\frac{1}{X^2} \qquad (12)$$

is to be replaced by

$$\lambda = -\ln\,(1-2m/X), \qquad (13)$$

then λ, X, m must <u>all</u> be declared in the ATØMIC statement; and λ and m must both be declared in DEPEND.

2. EPTERM statement. EPTERM must be defined, as in the program EINSTEIN, to tell the computer whether or not to truncate the answers at second order in EP. (cf. page 9.)

3. A "LBL PARAM" statement. This statement contains a list of the names of all atomic variables for which substitutions are to be made. For example, in the situation described above, λ appears

in the LBL PARAM statement. The entries in the LBL PARAM list must appear in precisely the same order as is used for the input data (cf. 4b below). The form of the LBL PARAM statement is as follows:

```
col.3 col.7
↓     ↓
LBL   PARAM   (NAME1, AA(1)),  (NAME2, AA(2)), ---
                 (NAMEN, AA(N)).
```

Here, NAME1, ---, NAMEN are the names of the atomic variables for which substitutions are being made. No more than 50 changes of variables can be made unless the user redimensions AA.

4. The data supplied by the user at the end of SUBSTITUTE is punched on cards with the first entry of each card in column 1, and the last entry before or in column 72. The data which must be supplied are as follows:

   a) Two integers, on a single card, giving the number of substitutions, "NSUB", to be made, and the number of complicated expressions, "NEXP", to be expanded with the substitutions. The first of these integers, "NSUB", must end in column 5 and the second, "NEXP", must end in column 10. In this card, and this card only, the data need not start in column 1.

   b) Next come the NSUB substitutions – i.e., expressions for the old atomic variables in terms of the new ones. These are read in one at a time, each preceded by a card listing a label or title of the quantity being read in, (these labels can be 72 characters in length), and each followed by a blank card. The substitution expressions must be read in with the same order as is used to list them in the LBL PARAM statement (see above). Each substitution expression can be many cards long – with each card's entries beginning in column 1 and ending before or in column 72. The computer identifies the end of a substitution expression by a dollar sign, $, with which the user must terminate each substitution.

c) The last quantities to be read in are the NEXP complicated expressions which are to be expanded in terms of the new atomic variables. These are read in one at a time, each preceded by a label card, each terminating in a dollar sign, and each followed by a blank card.

A listing of SUBSTITUTE, including sample data which illustrate the above instructions, follows.

2. A Complete Listing of SUBSTITUTE

```
$IBFMC ONE     NODECK
      SYMARG
C
C         PLACE ATOMIC AND DEPEND STATEMENTS FOLLOWING THIS COMMENT
C
        ATOMIC X,Y,Z,T,ALAM,ANU,EM,PHI   } User: Replace these cards
        DEPEND (ANU,ALAM,PHI,EM/X)       } with your own input.
C
C
        DIMENSION CARD(600),AA(50)
        DATA BLANK /6H      /
C
        INTEGER EPTERM
C
C         IF EPTERM = 0 TERMS CONTAINING ALL POWERS OF EXPANSION
C                       PARAMETER WILL BE RETAINED.
C         IF EPTERM .NE. 0 TERMS CONTAINING THE EXPANSION PARAMETER
C                          TO A POWER GREATER THAN 2 WILL BE DISCARDED.
C         A CARD CONTAINING THE INTEGER VALUE OF EPTERM MUST FOLLOW THIS
C         COMMENT.
C
        EPTERM = 0                       } User: Replace this card
C                                          with your own input.
      WRITE(6,202)
  202 FORMAT(57H1SUBSTITUTIONS TO BE MADE IN CURVATURE TENSOR EXPRESSIO
     1NS )
      WRITE(6,200)
  200 FORMAT(1H0)
      READ(5,101)NSUBS,NEXP
  101 FORMAT(2I5)
      DO 10 II = 1,NSUBS
      READ(5,102)(CARD(J),J=1,12)
  102 FORMAT(12A6)
      WRITE(6,105)(CARD(J),J=1,12)
  105 FORMAT(1H 5X 12A6)
      I = 0
    1 CONTINUE
       JJ = I+1
       JJJ = I+12
      READ(5,102)(CARD(J),J=JJ,JJJ)
       IF(CARD(JJ).EQ.BLANK)GO TO 3
      WRITE(6,106)(CARD(J),J=JJ,JJJ)
  106 FORMAT(10X,12A6)
      I = I + 12
        GO TO 1
    3 K = 0
       LET AA(II) = ALGCON CARD,K
   10 CONTINUE
C
C             PLACE THE LBL PARAM STATEMENT FOLLOWING THIS COMMENT
C
C
  LBL PARAM (ALAM,AA(1)), (ANU,AA(2))    } User: Replace this card
C                                          with your own input.
```

```
       DO 20 II = 1,NEXP
       WRITE(6,104)
 104  FORMAT(1H1)
       WRITE(6,200)
      READ(5,222)(CARD(J),J=1,12)
 222  FORMAT(12A6)
      WRITE(6,223)(CARD(J),J=1,12)
 223  FORMAT(5X,12A6)
      I = 0
  12  CONTINUE
      JJ = I + 1
      JJJ = I + 12
       READ(5,102)(CARD(J),J=JJ,JJJ)
       IF(CARD(JJ).EQ.BLANK)GO TO 2
       WRITE(6,105)(CARD(J),J=JJ,JJJ)
       I = I + 12
       GO TO 12
   2  K = 0
       LET EXP = ALGCON CARD,K
       LET ANS = SUBST EXP,LBL
       IF(EPTERM.EQ.0)GO TO 32
       LET ANS = EXPAND ANS
       ERASE EXP
       LET Z0 = COEFF ANS,EP**0,PP
       LET Z1 = COEFF ANS,EP**1,PP
       LET Z2 = COEFF ANS,EP**2,PP
       ERASE ANS
      LET EXP = EP**0*Z0 + EP**1*Z1 + EP**2*Z2
      GO TO 33
  32  CONTINUE
      LET EXP = ANS
  33  CONTINUE
       LET EXP = EXPAND EXP
      WRITE(6,107)
 107  FORMAT(34H0THE EXPRESSION AFTER SUBSTITUTION       )
       WRITE(6,200)
      S = 0.
  11  LET S = BCDCON EXP , CARD , 12
       WRITE(6,105)(CARD(J),J=2,12)
       IF(S.NE.0.)GO TO 11
  20  CONTINUE
      STOP
      END
```

Data for SUBSTITUTE

(User: Replace these data cards by your own data cards.)

```
     2     4
ALAM
- FMCLOG(1-2*EM/X)          $

ANU
2*PHI     $

1U    1L       EINSTEIN TENSOR COMPONENT
-X**(-2)*FMCEXP(-ALAM) + X**(-2) - X**(-1)*FMCEXP(-ALAM)*FMCDIF(ANU,X,1)
$

   2U    2L    EINSTEIN TENSOR COMPONENT
1./(X*2.)*FMCEXP(-ALAM) * FMCDIF(ALAM,X,1) - 1./(X*2.)
*FMCEXP(-ALAM)*FMCDIF(ANU,X,1)    + FMCEXP(-ALAM)*FMCDIF(ALAM,X,1)/4.
*FMCDIF(ANU,X,1) - FMCEXP(-ALAM)*FMCDIF(ANU,X,1)**2/4. - FMCEXP(-ALAM)
*FMCDIF(ANU,X,2)/2. $

   3U    3L    EINSTEIN TENSOR COMPONENT
1./(X*2.)*FMCEXP(-ALAM)*FMCDIF(ALAM,X,1) - 1./(X*2.)*FMCEXP(-ALAM)
*FMCDIF(ANU,X,1) + FMCEXP(-ALAM)*FMCDIF(ALAM,X,1)*FMCDIF(ANU,X,1)/4.
-FMCEXP(-ALAM)*FMCDIF(ANU,X,1)**2/4. -    FMCEXP(-ALAM)*FMCDIF(ANU,X,2)
/2. $

   4U    4L    EINSTEIN TENSOR COMPONENT
-1./(X*X)*FMCEXP(-ALAM) + 1./(X*X) + 1./X*FMCEXP(-ALAM)*FMCDIF(ALAM,X,1)
$
```

3. Output of SUBSTITUTE

The output for SUBSTITUTE begins with a reproduction of all the substitutions (changes of variable) to be made. Then follow, in the order in which they were input, the various complicated expressions to be expanded. These are printed, first in their original form, and immediately thereafter expanded in terms of the new atomic variables.

As an example, here is the output generated by the data which were given in the last section (NSUB = 2, NEXP = 4):

```
SUBSTITUTIONS TO BE MADE IN CURVATURE TENSOR EXPRESSIONS


      ALAM
           -FMCLOG(1-2*EM/X)          $
      ANU
          2*PHI     $






     1U   1L        EINSTEIN TENSOR COMPONENT
      -X**(-2)*FMCEXP(-ALAM) + X**(-2) - X**(-1)*FMCEXP(-ALAM)*FMCDIF(ANU,X,1)
      $

THE EXPRESSION AFTER SUBSTITUTION


      EM*X**(-3.0)*2.0+EM*X**(-2.0)*FMCDIF(PHI,(X,1))*4.0+X**(-1.0)*
      FMCDIF(PHI,(X,1))*(-2.0)$
      2U    2L    EINSTEIN TENSOR COMPONENT
     1./(X*2.)*FMCEXP(-ALAM) * FMCDIF(ALAM,X,1) - 1./(X*2.)
     *FMCEXP(-ALAM)*FMCDIF(ANU,X,1)    + FMCEXP(-ALAM)*FMCDIF(ALAM,X,1)/4.
     *FMCDIF(ANU,X,1) - FMCEXP(-ALAM)*FMCDIF(ANU,X,1)**2/4. - FMCEXP(-ALAM)
     *FMCDIF(ANU,X,2)/2. $

THE EXPRESSION AFTER SUBSTITUTION


     -EM*X**(-3.0)+EM*X**(-2.0)*FMCDIF(PHI,(X,1))+EM*X**(-1.0)*FMCDIF(
     PHI,(X,1))**2.0*2.0+EM*X**(-1.0)*FMCDIF(PHI,(X,2))*2.0+X**(-2.0)*
     FMCDIF(EM,(X,1))+X**(-1.0)*FMCDIF(EM,(X,1))*FMCDIF(PHI,(X,1))-X**(
     -1.0)*FMCDIF(PHI,(X,1))-FMCDIF(PHI,(X,1))**2.0-FMCDIF(PHI,(X,2))$
```

```
      3U    3L    EINSTEIN TENSOR COMPONENT
     1./(X*2.)*FMCEXP(-ALAM)*FMCDIF(ALAM,X,1) - 1./(X*2.)*FMCEXP(-ALAM)
     *FMCDIF(ANU,X,1) + FMCEXP(-ALAM)*FMCDIF(ALAM,X,1)*FMCDIF(ANU,X,1)/4.
     -FMCEXP(-ALAM)*FMCDIF(ANU,X,1)**2/4. -   FMCEXP(-ALAM)*FMCDIF(ANU,X,2)
     /2. $

THE EXPRESSION AFTER SUBSTITUTION


     -EM*X**(-3.0)+EM*X**(-2.0)*FMCDIF(PHI,(X,1))+EM*X**(-1.0)*FMCDIF(
     PHI,(X,1))**2.0*2.0+EM*X**(-1.0)*FMCDIF(PHI,(X,2))*2.0+X**(-2.0)*
     FMCDIF(EM,(X,1))+X**(-1.0)*FMCDIF(EM,(X,1))*FMCDIF(PHI,(X,1))-X**(
     -1.0)*FMCDIF(PHI,(X,1))-FMCDIF(PHI,(X,1))**2.0-FMCDIF(PHI,(X,2))$
```

```
      4U    4L    EINSTEIN TENSOR COMPONENT
     -1./(X*X)*FMCEXP(-ALAM) + 1./(X*X) + 1./X*FMCEXP(-ALAM)*FMCDIF(ALAM,X,1)
     $

THE EXPRESSION AFTER SUBSTITUTION


     X**(-2.0)*FMCDIF(EM,(X,1))*2.0$
```

4. Sample Problems for SUBSTITUTE

Input and output for one sample problem were given in the last two sections. As a second example we give input and output for a problem in which limited core storage actually forced us to use SUBSTITUTE:

Chitre, Hartle, and Thorne (1967) have been studying recently the fundamental modes of pulsation for slowly rotating, relativistic stellar models. To second order (ϵ^2) in the rotation and first order (ζ) in the pulsation, the metric for such a star can be put into the form

$$
\begin{aligned}
g_{11} &= e^{\lambda} \{1 + 2\,\epsilon^2\, r^{-1}\, e^{\lambda}(m_0 + m_2P_2) + 2\,\zeta r^{-1}\, e^{\lambda}\mu_0 - \epsilon^2\zeta(H_0 + H_2P_2)\} \\
g_{14} &= g_{41} = -\,\epsilon^2\zeta\; Q_2P_2 \\
g_{22} &= r^2\; \{1 + 2\,\epsilon^2\,(v_2 - h_2)P_2 - \epsilon^2\zeta\; K_2P_2\} \\
g_{33} &= r^2\, \sin^2\theta\; \{1 + 2\,\epsilon^2\,(v_2 - h_2)P_2 - \epsilon^2\zeta\; K_2P_2\} \\
g_{34} &= g_{43} = r^2\, \sin^2\theta\; \{\epsilon(\bar{\omega} - \Omega) - \epsilon\zeta j_1\} \\
g_{44} &= -\,e^{\nu}\; \{1 + 2\,\epsilon^2\,(h_0 + h_2P_2) - \epsilon^2\, e^{-\nu} r^2\, \sin^2\theta\; (\bar{\omega} - \Omega)^2 \\
&\qquad + \epsilon^2\zeta\;(N_0 + N_2P_2)\}
\end{aligned}
\tag{14}
$$

where $x^1 = r,\ x^2 = \theta,\ x^3 = \varphi,\ x^4 = t$, and

$$P_2 = \tfrac{1}{2}\,(3\,\cos^2\theta - 1). \tag{15}$$

In this metric λ, ν, $\bar{\omega}$, h_0, h_2, m_0, m_2, and v_2 are functions of r only; μ_0, j_1, H_0, H_2, N_0, N_2, and Q_2 are functions of r and t; and ϵ, ζ, Ω are constants.

This metric is so complicated that EINSTEIN was unable to calculate the corresponding mixed Christoffel symbols or Ricci tensor or Einstein tensor. An error number 7337 (computer's memory exhausted) was obtained when the computation was attempted.

To get around this difficulty, a simplified metric, equivalent to expression (14) but containing fewer terms, was put into EINSTEIN:

$$\begin{aligned}
g_{11} &= \mathcal{P}_k \{1 + \epsilon^2 \mathcal{B}\} , \\
g_{14} &= g_{41} = - \epsilon^2 \mathcal{F}_q , \\
g_{22} &= r^2 \{1 + \epsilon^2 \mathcal{C}\} , \\
g_{33} &= r^2 \sin^2\theta \{1 + \epsilon^2 \mathcal{C}\} , \\
g_{34} &= g_{43} = - \epsilon(r^2 \sin^2\theta) \mathcal{W} , \\
g_{44} &= - e^{\nu} \{1 + \epsilon^2 \mathcal{A}\} + \epsilon^2 (r^2 \sin^2\theta) \mathcal{W}^2 ;
\end{aligned} \tag{16a}$$

$$\begin{aligned}
g^{11} &= \mathcal{P}_e \{1 - \epsilon^2 \mathcal{B}\} , \\
g^{14} &= g^{41} = - \epsilon^2 e^{-\nu} \mathcal{P}_e \mathcal{F}_q , \\
g^{22} &= r^{-2} \{1 - \epsilon^2 \mathcal{C}\} , \\
g^{33} &= (r \sin\theta)^{-2} \{1 - \epsilon^2 \mathcal{C}\} - \epsilon^2 e^{-\nu} \mathcal{W}^2 , \\
g^{34} &= g^{43} = - \epsilon\, e^{-\nu} \mathcal{W} , \\
g^{44} &= - e^{-\nu} \{1 - \epsilon^2 \mathcal{A}\} ,
\end{aligned} \tag{16b}$$

Here the new variables $\mathcal{P}_k$, $\mathcal{P}_e$, and $\mathcal{W}$ are functions of r and t; while $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, and $\mathcal{F}_q$ are functions of r, θ, and t. EINSTEIN was able to compute the Ricci and Einstein tensors for this simplified metric to order ϵ^2 without exhausting the computer's memory. The resultant Ricci and Einstein tensors were then put through SUBSTITUTE, which made the following changes of variable – accurate to first order in ζ – to return them to the original notation:

$$
\begin{aligned}
\mathscr{P}_k &= e^{\lambda} \{1 + 2\,\zeta r^{-1} e^{\lambda} \mu_0\} , \\
\mathscr{P}_e &= e^{-\lambda} \{1 - 2\,\zeta r^{-1} e^{\lambda} \mu_0\} , \\
\mathscr{V} &= \Omega - \overline{\omega} + \zeta j_1 , \\
\mathscr{A} &= 2\,\zeta e^{-\nu} j_1 (\Omega - \overline{\omega}) r^2 \sin^2\theta + 2\, h_0 + \zeta N_0 \\
&\quad + (2\, h_2 + \zeta N_2)\, \tfrac{1}{2}\, (3 \cos^2\theta - 1) , \\
\mathscr{B} &= 2\, e^{\lambda} \{m_0/r + (m_2/r)\, \tfrac{1}{2}\, (3 \cos^2\theta - 1)\} \{1 - 2\,\zeta e^{\lambda} \mu_0/r\} \\
&\quad - \zeta\, \{H_0 + H_2\, \tfrac{1}{2}\, (3 \cos^2\theta - 1)\} , \\
\mathscr{C} &= \{2\, (v_2 - h_2) - \zeta K_2\}\, \tfrac{1}{2}\, (3 \cos^2\theta - 1) , \\
\mathscr{F}_q &= \zeta\, Q_2\, \tfrac{1}{2} (3 \cos^2\theta - 1) .
\end{aligned}
\qquad (17)
$$

In making these changes of variable by means of SUBSTITUTE, we exhausted the computer's memory on a few of the components of the Einstein tensor. When that happened, we broke the difficult components into pieces and performed the substitutions piece by piece.

As a guide to the user of SUBSTITUTE who has a long problem to do, we reproduce here the input to and the output from SUBSTITUTE for <u>part</u> of the above problem. Note that SUBSTITUTE does not truncate in powers of ζ (denoted EZ below), so that truncation must be done by hand.

INPUT

```
      ATOMIC X,Y,Z,T,EP,EZ
      ATOMIC ALAM,ANU,PIK,PIE,A,B,C,FQ,WR
      ATOMIC AMU0,AJ1,H0,H2,CH0,CH2,EN0,EN2,EM0,EM2,AK2,V2,Q2,OM,OMB
      DEPEND (ALAM,ANU,OMB,H0,H2,EM0,EM2,V2/X)
      DEPEND (PIK,PIE,WR,AMU0,AJ1,CH0,CH2,EN0,EN2,AK2,Q2/X,T)
      DEPEND (A,B,C,FQ/X,Y,T)
                                                      (§ III.E.1.1)

      EPTERM = 0                                      (§ III.E.1.2)

  LBL PARAM (PIK,AA(1)), (PIE,AA(2)), (WR,AA(3)), (A,AA(4)),
     1 (B,AA(5)), (C,AA(6)), (FQ,AA(7))               (§ III.E.1.3)



    7    2                                            (§ III.E.1.4 (a))
PIK =
FMCEXP(ALAM) * (1. + 2. * EZ *FMCEXP(ALAM) * AMU0/X)$

PIE =
FMCEXP(-ALAM) * (1.-2.*EZ*FMCEXP(ALAM)*AMU0/X)    $

WR =
OM - OMB + EZ * AJ1 $                                 (§ III.E.1.4 (b))

A =
2. * EZ * FMCEXP(-ANU) * AJ1 * (OM - OMB) * X * X * FMCSIN(Y) ** 2 + 2.
* H0 + EZ * EN0 + (2. * H2 + EZ * EN2) * 0.5 * (3. * FMCCOS(Y) **
2 - 1.) $

B =
2. * FMCEXP(ALAM) * ( EM0/X +  EM2/X * 0.5 * (3. * FMCCOS(Y) **2 - 1.))
* (1. - 2. * EZ * FMCEXP(ALAM) * AMU0/X) - EZ*(CH0+CH2* 0.5*
 (3.*FMCCOS(Y)**2 - 1.)) $

C =
(2. * (V2 - H2) - EZ * AK2) * 0.5 * (3. *FMCCOS(Y) ** 2 - 1.) $

FQ =
EZ * Q2 * 0.5 * (3. * FMCCOS(Y) ** 2 - 1.) $

1     3      RICCI TENSOR COMPONENT
5.0E-1*EP*PIE*X**2.0*FMCSIN(Y)**2.0*FMCEXP(-ANU)*FMCDIF(WR,(T,1),(
X,1))-2.5E-1*EP*PIE**2.0*X**2.0*FMCSIN(Y)**2.0*FMCEXP(-ANU)*FMCDIF
(PIK,(T,1))*FMCDIF(WR,(X,1))$
                                                      (§ III.E.1.4 (c))
4     3      RICCI TENSOR COMPONENT
-2.0*EP*PIE*X*FMCSIN(Y)**2.0*FMCEXP(-ANU)*FMCDIF(WR,(X,1))+
2.5E-1*EP*PIE*X**2.0*FMCSIN(Y)**2.0*FMCEXP(-ANU)*FMCDIF(ANU,(X,1))
*FMCDIF(WR,(X,1))-5.0E-1*EP*PIE*X**2.0*FMCSIN(Y)**2.0*FMCEXP(-ANU)
*FMCDIF(WR,(X,2))-2.5E-1*EP*PIE**2.0*X**2.0*FMCSIN(Y)**2.0*FMCEXP(
-ANU)*FMCDIF(PIK,(X,1))*FMCDIF(WR,(X,1))-5.0E-1*EP*X**2.0*FMCSIN(Y
)**2.0*FMCEXP(-ANU)*FMCDIF(PIE,(X,1))*FMCDIF(WR,(X,1))$
```

OUTPUT

```
      4    3        RICCI TENSOR COMPONENT
       -2.0*EP*PIE*X*FMCSIN(Y)**2.0*FMCEXP(-ANU)*FMCDIF(WR,(X,1))+
       2.5E-1*EP*PIE*X**2.0*FMCSIN(Y)**2.0*FMCEXP(-ANU)*FMCDIF(ANU,(X,1))
       *FMCDIF(WR,(X,1))-5.0E-1*EP*PIE*X**2.0*FMCSIN(Y)**2.0*FMCEXP(-ANU)
       *FMCDIF(WR,(X,2))-2.5E-1*EP*PIE**2.0*X**2.0*FMCSIN(Y)**2.0*FMCEXP(
       -ANU)*FMCDIF(PIK,(X,1))*FMCDIF(WR,(X,1))-5.0E-1*EP*X**2.0*FMCSIN(Y
       )**2.0*FMCEXP(-ANU)*FMCDIF(PIE,(X,1))*FMCDIF(WR,(X,1))$

THE EXPRESSION AFTER SUBSTITUTION


       AMU0*EP*EZ*X*FMCSIN(Y)**2.0*FMCEXP(-ANU)*FMCDIF(ANU,(X,1))*FMCDIF(
       OMB,(X,1))*5.0E-1-AMU0*EP*EZ*X*FMCSIN(Y)**2.0*FMCEXP(-ANU)*FMCDIF(
       OMB,(X,2))+AMU0*EP*EZ*FMCSIN(Y)**2.0*FMCEXP(-ANU)*FMCDIF(OMB,(X,1)
       )*(-3.49999998)+AMU0*EP*EZ**2.0*X*FMCSIN(Y)**2.0*FMCEXP(-ANU)*
       FMCDIF(AJ1,(X,1))*FMCDIF(ANU,(X,1))*(-5.0E-1)+AMU0*EP*EZ**2.0*X*
       FMCSIN(Y)**2.0*FMCEXP(-ANU)*FMCDIF(AJ1,(X,2))+AMU0*EP*EZ**2.0*
       FMCSIN(Y)**2.0*FMCEXP(ALAM-ANU)*FMCDIF(AMU0,(X,1))*FMCDIF(OMB,(X,1
       ))*(-2.0)+AMU0*EP*EZ**2.0*FMCSIN(Y)**2.0*FMCEXP(-ANU)*FMCDIF(AJ1,(
       X,1))*3.49999998+AMU0*EP*EZ**3.0*FMCSIN(Y)**2.0*FMCEXP(ALAM-ANU)*
       FMCDIF(AJ1,(X,1))*FMCDIF(AMU0,(X,1))*2.0+AMU0**2.0*EP*EZ**2.0*X**(
       -1.0)*FMCSIN(Y)**2.0*FMCEXP(ALAM-ANU)*FMCDIF(OMB,(X,1))*2.0+AMU0**
       2.0*EP*EZ**2.0*FMCSIN(Y)**2.0*FMCEXP(ALAM-ANU)*FMCDIF(ALAM,(X,1))*
       FMCDIF(OMB,(X,1))*(-3.0)+AMU0**2.0*EP*EZ**3.0*X**(-1.0)*FMCSIN(Y)
       **2.0*FMCEXP(ALAM-ANU)*FMCDIF(AJ1,(X,1))*(-2.0)+AMU0**2.0*EP*EZ**
       3.0*X**(-1.0)*FMCSIN(Y)**2.0*FMCEXP(ALAM*2.0-ANU)*FMCDIF(AMU0,(X,1
       ))*FMCDIF(OMB,(X,1))*2.0+AMU0**2.0*EP*EZ**3.0*FMCSIN(Y)**2.0*
       FMCEXP(ALAM-ANU)*FMCDIF(AJ1,(X,1))*FMCDIF(ALAM,(X,1))*3.0+AMU0**
       2.0*EP*EZ**4.0*X**(-1.0)*FMCSIN(Y)**2.0*FMCEXP(ALAM*2.0-ANU)*
       FMCDIF(AJ1,(X,1))*FMCDIF(AMU0,(X,1))*(-2.0)+AMU0**3.0*EP*EZ**3.0*X
       **(-2.0)*FMCSIN(Y)**2.0*FMCEXP(ALAM*2.0-ANU)*FMCDIF(OMB,(X,1))*(
       -2.0)+AMU0**3.0*EP*EZ**3.0*X**(-1.0)*FMCSIN(Y)**2.0*FMCEXP(ALAM*
       2.0-ANU)*FMCDIF(ALAM,(X,1))*FMCDIF(OMB,(X,1))*4.0+AMU0**3.0*EP*EZ
       **4.0*X**(-2.0)*FMCSIN(Y)**2.0*FMCEXP(ALAM*2.0-ANU)*FMCDIF(AJ1,(X,
       1))*2.0+AMU0**3.0*EP*EZ**4.0*X**(-1.0)*FMCSIN(Y)**2.0*FMCEXP(ALAM*
       2.0-ANU)*FMCDIF(AJ1,(X,1))*FMCDIF(ALAM,(X,1))*(-4.0)+EP*EZ*X*
       FMCSIN(Y)**2.0*FMCEXP(-ALAM-ANU)*FMCDIF(AJ1,(X,1))*(-2.0)+EP*EZ*X*
       FMCSIN(Y)**2.0*FMCEXP(-ANU)*FMCDIF(AMU0,(X,1))*FMCDIF(OMB,(X,1))*(
       -5.0E-1)+EP*EZ*X**2.0*FMCSIN(Y)**2.0*FMCEXP(-ALAM-ANU)*FMCDIF(AJ1,
       (X,1))*FMCDIF(ALAM,(X,1))*2.5E-1+EP*EZ*X**2.0*FMCSIN(Y)**2.0*
       FMCEXP(-ALAM-ANU)*FMCDIF(AJ1,(X,1))*FMCDIF(ANU,(X,1))*2.5E-1+EP*EZ
       *X**2.0*FMCSIN(Y)**2.0*FMCEXP(-ALAM-ANU)*FMCDIF(AJ1,(X,2))*(
       -5.0E-1)+EP*EZ**2.0*X*FMCSIN(Y)**2.0*FMCEXP(-ANU)*FMCDIF(AJ1,(X,1)
       )*FMCDIF(AMU0,(X,1))*5.0E-1+EP*X*FMCSIN(Y)**2.0*FMCEXP(-ALAM-ANU)*
       FMCDIF(OMB,(X,1))*2.0+EP*X**2.0*FMCSIN(Y)**2.0*FMCEXP(-ALAM-ANU)*
       FMCDIF(ALAM,(X,1))*FMCDIF(OMB,(X,1))*(-2.5E-1)+EP*X**2.0*FMCSIN(Y)
       **2.0*FMCEXP(-ALAM-ANU)*FMCDIF(ANU,(X,1))*FMCDIF(OMB,(X,1))*(
       -2.5E-1)+EP*X**2.0*FMCSIN(Y)**2.0*FMCEXP(-ALAM-ANU)*FMCDIF(OMB,(X,
       2))*5.0E-1$
```

```
1    3       RICCI TENSOR COMPONENT
 5.0E-1*EP*PIE*X**2.0*FMCSIN(Y)**2.0*FMCEXP(-ANU)*FMCDIF(WR,(T,1),(
 X,1))-2.5E-1*EP*PIE**2.0*X**2.0*FMCSIN(Y)**2.0*FMCEXP(-ANU)*FMCDIF
 (PIK,(T,1))*FMCDIF(WR,(X,1))$

THE EXPRESSION AFTER SUBSTITUTION


 -AMUO*EP*EZ**2.0*X*FMCSIN(Y)**2.0*FMCEXP(-ANU)*FMCDIF(AJ1,(T,1),(X
 ,1))+AMUO*EP*EZ**2.0*FMCSIN(Y)**2.0*FMCEXP(ALAM-ANU)*FMCDIF(AMUO,(
 T,1))*FMCDIF(OMB,(X,1))*(-2.0)+AMUO*EP*EZ**3.0*FMCSIN(Y)**2.0*
 FMCEXP(ALAM-ANU)*FMCDIF(AJ1,(X,1))*FMCDIF(AMUO,(T,1))*2.0+AMUO**
 2.0*EP*EZ**3.0*X**(-1.0)*FMCSIN(Y)**2.0*FMCEXP(ALAM*2.0-ANU)*
 FMCDIF(AMUO,(T,1))*FMCDIF(OMB,(X,1))*2.0+AMUO**2.0*EP*EZ**4.0*X**(
 -1.0)*FMCSIN(Y)**2.0*FMCEXP(ALAM*2.0-ANU)*FMCDIF(AJ1,(X,1))*FMCDIF
 (AMUO,(T,1))*(-2.0)+EP*EZ*X*FMCSIN(Y)**2.0*FMCEXP(-ANU)*FMCDIF(
 AMUO,(T,1))*FMCDIF(OMB,(X,1))*5.0E-1+EP*EZ*X**2.0*FMCSIN(Y)**2.0*
 FMCEXP(-ALAM-ANU)*FMCDIF(AJ1,(T,1),(X,1))*5.0E-1+EP*EZ**2.0*X*
 FMCSIN(Y)**2.0*FMCEXP(-ANU)*FMCDIF(AJ1,(X,1))*FMCCIF(AMUO,(T,1))*(
 -5.0E-1)$
```

IV. A BRIEF GUIDE TO FORMAC PROGRAMMING*

A. The FORMAC System

In writing a program using FORMAC it is helpful to consider the FORMAC compiler as an extension of the facilities of FORTRAN IV which allows for the manipulation of algebraic symbols. The FORTRAN IV features, such as integer and floating point arithmatic, type statements, data statements, and logical operations, all exist and can be used in FORMAC programs.

When a FORMAC program is submitted for compilation the machine immediately produces a purely FORTRAN program, and the FORMAC commands are rendered as comment statements.

*For further details see the FORMAC manual (IBM 1962).

B. FORMAC Executable Statements

Because algebraic quantities rather than numbers are usually being manipulated by a FORMAC program, input data are considered as alphanumeric information and are read into the machine by the FORTRAN IV "A" format. A thirty-six bit computer word in core storage can contain 6 alphanumeric characters. Thus 72 columns on an IBM data card represent 12 computer words. The program SUBSTITUTE illustrates how an algebraic expression up to 50 cards in length can be read into memory.

Once an expression, say ALGEXP, has been entered, it must be converted to a form that can be manipulated conveniently by the compiler. This is done by the FORMAC executable statement ALGCØN

```
K = 0
LET A = ALGCØN  ALGEXP,K
```

It must be emphasized that all variables which appear in expressions that are read in as data must be declared in the ATØMIC and DEPEND statements.

When a quantity, such as A above, has been put into proper form by means of an ALGCØN statement, it can then be operated on by other FORMAC executable statements. The following executable statements are available in FORMAC. (Note that some of these statements have been described in earlier sections.)

AUTSIM	Used to evaluate elementary functions that have constants or FORTRAN-type expressions as arguments.

BCDCØN	To be described below.

CENSUS	Used to count the number of computer words, terms, or factors in a FORMAC expression.

CØEFF Used to obtain the coefficient, in an algebraic expression, of an atomic variable or an atomic variable raised to a specified power.

ERASE Used to erase a FORMAC quantity in core storage.

EVAL Used to evaluate a FORMAC expression using supplied values of the variables. This results in a numeric quantity.

EXPAND Used to simplify, collect terms, and remove parentheses in algebraic expressions.

FIND Used to establish a function relation between a FORMAC expression and prescribed atomic variables.

LET Used to define algebraic expressions.

MATCH See below.

PART See FORMAC manual.

SUBST See below.

ØRDER Used to specify the sequence in which factors in products, and terms in sums are to appear in expressions.

These comman , are described in more detail in the FORMAC manual, and most of them were used in the ALBERT programs.

It is useful to note that most operations performed on numeric quantities have their FORMAC analogs. For example, logical operations can be performed on FORMAC variables just as they can be on FORTRAN variables: One algebraic expression can be compared to another by the FORMAC function MATCH. The sequence

```
ATOMIC X, Y
LOGICAL Q
```

```
LET C = X**2 + Y**2
LET Q = MATCH ID, B, C
IF(Q) GO TO 2
CØNTINUE
```

would compare B to C; and if they were equivalent, it would set the value of Q to the FORTRAN logical TRUE and then transfer to statement number 2 for its next instruction.

Once a desired algebraic quantity has been constructed, it can be evaluated numerically, or substitutions can be made that will convert it to yet another symbolic quantity. The FORMAC operation

```
LET Z = SUBST B, LIST
```

will result in a new algebraic quantity, Z. LIST is a listing of the symbols in B and the names of new expressions that are to take their places. The actual formation of LIST is done by the FORMAC function, PARAM. For example, consider the following:

```
      ATØMIC X, Y          ALPHA, BETA
      LET C = X**2 + Y**2
      LET A = ALPHA**2
      LET B = BETA**2
LIST  PARAM (X,A), (Y,B)
      LET Z = SUBST C, LIST
```

Z now has the value ALPHA**4 + BETA**4. Notice that the statements on cards begin in column 7 but the word, LIST, is placed in columns 1 → 4.

Assuming a computation has been completed, the information, other than numeric, must be put into a form suitable for printing. This is accomplished by the FORMAC command BCDCØN.

```
DIMENSIØN CARD (12)
ANS = 0
LET ANS = BCDCØN Z, CARD,12
```

will cause alphanumeric information to be stored in the 12 computer words of the array CARD. This can in turn be written out in FORTRAN's "A" format.

C. Ways to Avoid Exhausting the Computer's Memory

What has been attempted with the ALBERT programs is to make them as simple and easy to use as possible. As mentioned earlier, FORMAC on the IBM 7094 suffers from a severe lack of core storage. The program EINSTEIN plus system overhead took in the neighborhood of 5/7 of the 32K memory capacity of the machine. Reading the input metrics in on data cards would conserve some core space; but for clarity we chose to define the metric by means of LET statements in the body of the program. We have circumvented the core problem to some extent by simplification of the metrics themselves, by punching the output from EINSTEIN on cards, and by expanding that output in terms of new variables using the SUBSTITUTION program (cf. § III.E.4).

FORMAC provides for the use of an overlay feature, which might be another alternative for solving core problems. Yet another, is to purge from the system portions of the user's program after they have been executed (see Clemens and Matzner 1967). However, the only truly satisfactory solution to the core problem will be to go to a computer with a larger memory — e.g. the IBM 360 series.

V. REFERENCES

Chitre, A. M., Hartle, J. B., and Thorne, K. S. 1967, "The Fundamental Mode of Pulsation of Slowly Rotating, Relativistic Stellar Models," in preparation.

Clemens, R. and Matzner, R. 1967, "A System for Symbolic Computation of the Riemann Tensor," Technical Report No. 635, University of Maryland, Department of Physics and Astronomy, College Park, Maryland.

Fletcher, J. G. 1965, "Grad Assistant – A Program for Symbolic Algebraic Manipulation and Differentiation," UCRL Report 14624-T.

____________ 1966, "Grad Assistant – A Program for Symbolic Algebraic Manipulation and Differentiation," Communications of Assoc. for Computing Machinery, **9**, 552.

Fletcher, J. G., Clemens, R., Matzner, R., Thorne, K. S., and Zimmerman, B. A. 1967, "Computer Programs for Calculating General Relativistic Curvature Tensors," Ap. J. Letters, **148**, L91.

Hartle, J. B. 1967, "Slowly Rotating Relativistic Stars, I. Equations of Structure," Ap. J., to be published December 1967.

IBM 1964, User's Preliminary Reference Manual for the Experimental FORmula MAnipulation Compiler (FORMAC), (International Business Machines Corporation, Boston Advanced Programming, 545 Technology Square (3rd floor), Cambridge, Massachusetts 02139).

Thorne, K. S. and Campolattaro, A. 1967, "Nonradial Pulsation of General-Relativistic Stellar Models: I – Analytic Analysis for $l \geq 2$," Ap. J., to be published September 1967.